

Toshihiko Fukuoka et al
January 16, 2002

日本国特許庁

JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日

Date of Application:

2001年 1月22日

出願番号

Application Number:

特願2001-013632

出願人

Applicant(s):

松下電器産業株式会社

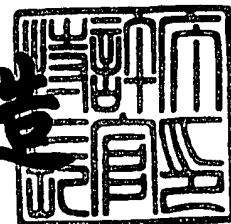
jc997 U.S. PTO
10/046263
01/16/02

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年11月 2日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3097

Docket No.: 60188-144

RS
2
2-27-02
PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of :
Toshihiko FUKUOKA, et al. :
Serial No.: : Group Art Unit:
Filed: January 16, 2002 : Examiner:
For: DEVICE AND METHOD FOR ERROR AND SYNC DETECTION

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Commissioner for Patents
Washington, DC 20231

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claims the priority
of:

Japanese Patent Application Number 2001-013632, Filed January 22, 2001

cited in the Declaration of the present application. A Certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:kjw
Date: January 16, 2002
Facsimile: (202) 756-8087

【書類名】 特許願

【整理番号】 2037820075

【提出日】 平成13年 1月22日

【あて先】 特許庁長官 殿

【国際特許分類】 H03M 13/00

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 福岡 俊彦

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 和田 妙美

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100077931

【弁理士】

【氏名又は名称】 前田 弘

【選任した代理人】

【識別番号】 100094134

【弁理士】

【氏名又は名称】 小山 廣毅

【選任した代理人】

【識別番号】 100110939

【弁理士】

【氏名又は名称】 竹内 宏

【選任した代理人】

【識別番号】 100110940

【弁理士】

【氏名又は名称】 嶋田 高久

【選任した代理人】

【識別番号】 100113262

【弁理士】

【氏名又は名称】 竹内 祐二

【選任した代理人】

【識別番号】 100115059

【弁理士】

【氏名又は名称】 今江 克実

【選任した代理人】

【識別番号】 100115510

【弁理士】

【氏名又は名称】 手島 勝

【選任した代理人】

【識別番号】 100115691

【弁理士】

【氏名又は名称】 藤田 篤史

【手数料の表示】

【予納台帳番号】 014409

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0006010

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 誤り及び同期検出装置並びに方法

【特許請求の範囲】

【請求項 1】 7ビット単位のバイトデータを入力して8ビット単位のバイトデータに変換するデータ並び替えブロックと、

前記データ並び替えブロックで変換された8ビット単位のバイトデータを入力し、該バイトデータを用いてMPEG同期バイトの検出及びパリティチェックによる誤りの検出を行うパリティチェックブロックと、

前記データ並び替えブロックで変換された8ビット単位のバイトデータ、及び前記パリティチェックブロックにおけるMPEG同期バイトの検出及びパリティチェックによる誤り検出のための演算途中の8ビット単位の中間バイトデータを入力して記憶し、8ビット単位のバイトデータを基本単位としてデータの入出力が可能なデータ記憶ブロックとを備えて、

前記パリティチェックブロックから、同期バイトを含む8ビット単位のバイトデータの集合であるMPEGパケットデータを出力する

ことを特徴とする誤り及び同期検出装置。

【請求項 2】 前記データ並び替えブロックは、

連続して入力される7ビット単位の2つのバイトデータを1組として、最初に入力されるバイトデータと後に入力されるバイトデータの上位1ビットとを組み合わせた8ビット単位の第1のバイトデータと、

前記最初に入力されるバイトデータの下位6ビットと後に入力されるバイトデータの上位2ビットとを組み合わせた8ビット単位の第2のバイトデータと、

同様に最初に入力されるバイトデータの下位 n ビット ($n=5, 4, 3, 2, 1$) と後に入力されるバイトデータの上位 m ビット ($m=3, 4, 5, 6, 7$) とを組み合わせた8ビット単位の第3～第7のバイトデータとを生成する

ことを特徴とする請求項 1 記載の誤り及び同期検出装置。

【請求項 3】 前記パリティチェックブロックは、

データの所定クロック分の遅延を含む所定のシンドローム計算を行う第1及び第2の演算ブロックを有し、

前記第 1 の演算ブロックは、前記データ並び替えブロックから出力される 8 ビット単位のバイトデータを受けて、前記データの所定クロック分の遅延前の演算を行い、この演算結果である中間バイトデータを前記データ記憶ブロックに出力し、

前記第 2 の演算ブロックは、前記データ記憶ブロックから前記中間バイトデータを受けて、前記データの所定クロック分の遅延前の演算を行って、MPEG 同期検出及びパリティチェックによる誤り検出を行った 8 ビット単位のバイトデータを出力する

ことを特徴とする請求項 1 記載の誤り及び同期検出装置。

【請求項 4】 前記第 1 の演算ブロックは、

入力される 8 ビット単位のバイトデータを $pdatai[7:0]$ 、このバイトデータ $pdatai[7:0]$ を用いて行った演算結果を示すデータを $gxot[7:0]$ 、この演算結果データ $gxot[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $gxot7d[7:0]$ 、前記演算結果データ $gxot[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $gx[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、該 8 ビット単位の間変数 $gx[7:0]$ を構成する各ビット $gx[7]$ 、 $gx[6]$ 、 $gx[5]$ 、 $gx[4]$ 、 $gx[3]$ 、 $gx[2]$ 、 $gx[1]$ 、 $gx[0]$ を、各々、

$$gx[0] = gxot7d[0]$$

$$gx[1] = gxot7d[1]$$

$$gx[2] = gxot7d[2] \wedge gxot7d[0]$$

$$gx[3] = gxot7d[3] \wedge gxot7d[1] \wedge gxot7d[0]$$

$$gx[4] = gxot7d[4] \wedge gxot7d[2] \wedge gxot7d[1]$$

$$gx[5] = gxot7d[5] \wedge gxot7d[3] \wedge gxot7d[2]$$

$$gx[6] = gxot7d[6] \wedge gxot7d[4] \wedge gxot7d[3]$$

$$gx[7] = gxot7d[7] \wedge gxot7d[5] \wedge gxot7d[4] \wedge gxot7d[0]$$

の演算により算出し、

前記演算結果データ $gxot[7:0]$ を構成する各ビット $gxot[7]$ 、 $gxot[6]$ 、 $gxot[5]$ 、 $gxot[4]$ 、 $gxot[3]$ 、 $gxot[2]$ 、 $gxot[1]$ 、 $gxot[0]$ を、前記中間変数 $gx[7:0]$ を用いて、各々、

```

gxot[7] = gx[7] ^ pdatai[7]
gxot[6] = gx[7] ^ gx[6] ^ pdatai[7] ^ pdatai[6]
gxot[5] = gx[7] ^ gx[6] ^ gx[5]
          ^ pdatai[7] ^ pdatai[6] ^ pdatai[5]
gxot[4] = gx[7] ^ gx[6] ^ gx[5] ^ gx[4]
          ^ pdatai[7] ^ pdatai[6] ^ pdatai[5] ^ pdatai[4]
gxot[3] = gx[7] ^ gx[6] ^ gx[5] ^ gx[4] ^ gx[3]
          ^ pdatai[7] ^ pdatai[6] ^ pdatai[5] ^ pdatai[4] ^ pdatai[3]
gxot[2] = gx[6] ^ gx[5] ^ gx[4] ^ gx[3] ^ gx[2]
          ^ pdatai[6] ^ pdatai[5] ^ pdatai[4] ^ pdatai[3] ^ pdatai[2]
gxot[1] = gx[5] ^ gx[4] ^ gx[3] ^ gx[2] ^ gx[1]
          ^ pdatai[5] ^ pdatai[4] ^ pdatai[3] ^ pdatai[2] ^ pdatai[1]
gxot[0] = gx[4] ^ gx[3] ^ gx[2] ^ gx[1] ^ gx[0]
          ^ pdatai[4] ^ pdatai[3] ^ pdatai[2] ^ pdatai[1] ^ pdatai[0]

```

の演算により算出する

ことを特徴とする請求項 3 記載の誤り及び同期検出装置。

【請求項 5】 前記第 2 の演算ブロックは、

入力される 8 ビット単位のバイトデータを $\text{dobx}[7:0]$ 、このバイトデータ $\text{dobx}[7:0]$ を用いて行った演算の結果を示すデータを $\text{bxot1}[7:0]$ 、前記入力されるバイトデータ $\text{dobx}[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $\text{dobx7d}[7:0]$ 、前記演算結果データ $\text{bxot1}[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $\text{bx}[7:0]$ 、前記第 1 の演算ブロックの演算結果データ $\text{gxot}[7:0]$ を前記基準クロックに従って 1 クロック分遅延させた 8 ビット単位のバイトデータを $\text{gxot1d}[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、前記 8 ビット単位の間変数 $\text{bx}[7:0]$ を構成する各ビット $\text{bx}[7]$ 、 $\text{bx}[6]$ 、 $\text{bx}[5]$ 、 $\text{bx}[4]$ 、 $\text{bx}[3]$ 、 $\text{bx}[2]$ 、 $\text{bx}[1]$ 、 $\text{bx}[0]$ を、各々、

$\text{bx}[0] = \text{dobx7d}[0]$

$\text{bx}[1] = \text{dobx7d}[1]$

$\text{bx}[2] = \text{dobx7d}[2]$

$$\begin{aligned} \text{bx}[3] &= \text{dobx7d}[3] \\ \text{bx}[4] &= \text{dobx7d}[4] \\ \text{bx}[5] &= \text{dobx7d}[5] \wedge \text{dobx}[1] \\ \text{bx}[6] &= \text{dobx7d}[6] \wedge \text{dobx}[2] \\ \text{bx}[7] &= \text{dobx7d}[7] \wedge \text{dobx}[3] \wedge \text{dobx}[1] \end{aligned}$$

の演算で算出し、

また、前記演算結果データ $\text{bxot1}[7:0]$ を構成する各ビット $\text{bxot1}[7]$ 、 $\text{bxot1}[6]$ 、 $\text{bxot1}[5]$ 、 $\text{bxot1}[4]$ 、 $\text{bxot1}[3]$ 、 $\text{bxot1}[2]$ 、 $\text{bxot1}[1]$ 、 $\text{bxot1}[0]$ を、各々、

$$\begin{aligned} \text{bxot1}[7] &= \text{bx}[7] \wedge \text{dobx}[0] \\ \text{bxot1}[6] &= \text{bx}[6] \wedge \text{bx}[0] \wedge \text{dobx}[7] \\ \text{bxot1}[5] &= \text{bx}[5] \wedge \text{dobx}[7] \wedge \text{dobx}[6] \\ \text{bxot1}[4] &= \text{bx}[4] \wedge \text{bx}[0] \wedge \text{dobx}[6] \wedge \text{dobx}[5] \\ \text{bxot1}[3] &= \text{bx}[3] \wedge \text{dobx}[7] \wedge \text{dobx}[5] \wedge \text{dobx}[4] \\ \text{bxot1}[2] &= \text{bx}[2] \wedge \text{dobx}[6] \wedge \text{dobx}[4] \wedge \text{dobx}[3] \\ \text{bxot1}[1] &= \text{bx}[1] \wedge \text{dobx}[5] \wedge \text{dobx}[3] \wedge \text{dobx}[2] \\ \text{bxot1}[0] &= \text{bx}[0] \wedge \text{dobx}[4] \wedge \text{dobx}[2] \wedge \text{dobx}[1] \end{aligned}$$

の演算で算出し、

また、前記演算結果データ $\text{bxot1}[7:0]$ 及び前記第 1 の演算ブロックの演算結果データ $\text{gxot}[7:0]$ に基づいて行った演算の結果を示す 8 ビット単位のバイトデータ $\text{bxot2}[7:0]$ を、

$$\text{bxot2}[7:0] = \text{bxot1}[7:0] \wedge \text{gxot1d}[7:0]$$

の演算で算出する

ことを特徴とする請求項 3 記載の誤り及び同期検出装置。

【請求項 6】 前記データ記憶ブロックは、

前記データ並び替えブロックからの 8 ビット単位のバイトデータ、及び前記第 1 の演算ブロックの演算結果である 8 ビット単位のバイトデータを入力し、この両バイトデータを各々特定の期間保持した後に出力する

ことを特徴とする請求項 3 記載の誤り及び同期検出装置。

【請求項 7】 前記データ記憶ブロックは RAM により構成される

ことを特徴とする請求項 1 又は 6 記載の誤り及び同期検出装置。

【請求項 8】 7 ビット単位のバイトデータを入力して 8 ビット単位のバイトデータに変換するデータ並び替え工程と、

前記データ並び替えブロックで変換された 8 ビット単位のバイトデータを入力し、該バイトデータを用いてシンドローム計算し、その計算途中の 8 ビット単位の間バイトデータをデータ記憶装置に一時記憶させながら、該データ記憶装置に記憶した前記中間バイトデータを用いて前記シンドローム計算を続行して、MPEG 同期バイトの検出及びパリティチェックによる誤りの検出を行うパリティチェック工程とを備え、

同期バイトを含む 8 ビット単位のバイトデータの集合である MPEG パケットデータを出力する

ことを特徴とする誤り及び同期検出方法。

【請求項 9】 前記データ並び替え工程では、

連続して入力される 7 ビット単位の 2 つのバイトデータを 1 組として、最初に入力されるバイトデータと後に入力されるバイトデータの上位 1 ビットとを組み合わせた 8 ビット単位の第 1 のバイトデータと、

前記最初に入力されるバイトデータの下位 6 ビットと後に入力されるバイトデータの上位 2 ビットとを組み合わせた 8 ビット単位の第 2 のバイトデータと、

同様に最初に入力されるバイトデータの下位 n ビット ($n = 5, 4, 3, 2, 1$) と後に入力されるバイトデータの上位 m ビット ($m = 3, 4, 5, 6, 7$) とを組み合わせた 8 ビット単位の第 3 ～ 第 7 のバイトデータとを生成する

ことを特徴とする請求項 8 記載の誤り及び同期検出方法。

【請求項 10】 前記パリティチェック工程は、

データの所定クロック分の遅延を含む所定のシンドローム計算を行う第 1 及び第 2 の演算工程を有し、

前記第 1 の演算工程では、前記データ並び替え工程で変換された 8 ビット単位のバイトデータを受けて、前記データの所定クロック分の遅延前の演算を行い、この演算結果である 8 ビット単位の間バイトデータを前記データ記憶装置に出力し、

前記第 2 の演算工程では、前記データ記憶装置から前記中間バイトデータを受けて、前記データの所定クロック分の遅延前の演算を行って、MPEG 同期検出及びパリティチェックによる誤り検出を行った 8 ビット単位のバイトデータを出力する

ことを特徴とする請求項 8 記載の誤り及び同期検出方法。

【請求項 11】 前記第 1 の演算工程では、

入力される 8 ビット単位のバイトデータを $pdatai[7:0]$ 、このバイトデータ $pdatai[7:0]$ を用いて行った演算結果を示すデータを $gxot[7:0]$ 、この演算結果データ $gxot[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $gxot7d[7:0]$ 、前記演算結果データ $gxot[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $gx[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、該 8 ビット単位の間変数 $gx[7:0]$ を構成する各ビット $gx[7]$ 、 $gx[6]$ 、 $gx[5]$ 、 $gx[4]$ 、 $gx[3]$ 、 $gx[2]$ 、 $gx[1]$ 、 $gx[0]$ を、各々、

$$gx[0] = gxot7d[0]$$

$$gx[1] = gxot7d[1]$$

$$gx[2] = gxot7d[2] \wedge gxot7d[0]$$

$$gx[3] = gxot7d[3] \wedge gxot7d[1] \wedge gxot7d[0]$$

$$gx[4] = gxot7d[4] \wedge gxot7d[2] \wedge gxot7d[1]$$

$$gx[5] = gxot7d[5] \wedge gxot7d[3] \wedge gxot7d[2]$$

$$gx[6] = gxot7d[6] \wedge gxot7d[4] \wedge gxot7d[3]$$

$$gx[7] = gxot7d[7] \wedge gxot7d[5] \wedge gxot7d[4] \wedge gxot7d[0]$$

の演算により算出し、

前記演算結果データ $gxot[7:0]$ を構成する各ビット $gxot[7]$ 、 $gxot[6]$ 、 $gxot[5]$ 、 $gxot[4]$ 、 $gxot[3]$ 、 $gxot[2]$ 、 $gxot[1]$ 、 $gxot[0]$ を、前記中間変数 $gx[7:0]$ を用いて、各々、

$$gxot[7] = gx[7] \wedge pdatai[7]$$

$$gxot[6] = gx[7] \wedge gx[6] \wedge pdatai[7] \wedge pdatai[6]$$

$$gxot[5] = gx[7] \wedge gx[6] \wedge gx[5]$$

$$\wedge pdatai[7] \wedge pdatai[6] \wedge pdatai[5]$$

$$\begin{aligned}
 \text{gxot}[4] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \\
 &\quad \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \\
 \text{gxot}[3] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \\
 &\quad \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \\
 \text{gxot}[2] &= \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \\
 &\quad \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \\
 \text{gxot}[1] &= \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \\
 &\quad \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \\
 \text{gxot}[0] &= \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \wedge \text{gx}[0] \\
 &\quad \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \wedge \text{pdatai}[0]
 \end{aligned}$$

の演算により算出する

ことを特徴とする請求項 1 0 記載の誤り及び同期検出方法。

【請求項 1 2】 前記第 2 の演算工程は、

入力される 8 ビット単位のバイトデータを $\text{dobx}[7:0]$ 、このバイトデータ $\text{dobx}[7:0]$ を用いて行った演算の結果を示すデータを $\text{bxot1}[7:0]$ 、前記入力されるバイトデータ $\text{dobx}[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $\text{dobx7d}[7:0]$ 、前記演算結果データ $\text{bxot1}[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $\text{bx}[7:0]$ 、前記第 1 の演算ブロックの演算結果データ $\text{gxot1}[7:0]$ を前記基準クロックに従って 1 クロック分遅延させた 8 ビット単位のバイトデータを $\text{gxot1d}[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、前記 8 ビット単位の間変数 $\text{bx}[7:0]$ を構成する各ビット $\text{bx}[7]$ 、 $\text{bx}[6]$ 、 $\text{bx}[5]$ 、 $\text{bx}[4]$ 、 $\text{bx}[3]$ 、 $\text{bx}[2]$ 、 $\text{bx}[1]$ 、 $\text{bx}[0]$ を、各々、

$$\begin{aligned}
 \text{bx}[0] &= \text{dobx7d}[0] \\
 \text{bx}[1] &= \text{dobx7d}[1] \\
 \text{bx}[2] &= \text{dobx7d}[2] \\
 \text{bx}[3] &= \text{dobx7d}[3] \\
 \text{bx}[4] &= \text{dobx7d}[4] \\
 \text{bx}[5] &= \text{dobx7d}[5] \wedge \text{dobx}[1] \\
 \text{bx}[6] &= \text{dobx7d}[6] \wedge \text{dobx}[2]
 \end{aligned}$$

$$bx[7] = dobx7d[7] \wedge dobx[3] \wedge dobx[1]$$

の演算で算出し、

また、前記演算結果データ $bxot1[7:0]$ を構成する各ビット $bxot1[7]$ 、 $bxot1[6]$ 、 $bxot1[5]$ 、 $bxot1[4]$ 、 $bxot1[3]$ 、 $bxot1[2]$ 、 $bxot1[1]$ 、 $bxot1[0]$ を、各々、

$$bxot1[7] = bx[7] \wedge dobx[0]$$

$$bxot1[6] = bx[6] \wedge bx[0] \wedge dobx[7]$$

$$bxot1[5] = bx[5] \wedge dobx[7] \wedge dobx[6]$$

$$bxot1[4] = bx[4] \wedge bx[0] \wedge dobx[6] \wedge dobx[5]$$

$$bxot1[3] = bx[3] \wedge dobx[7] \wedge dobx[5] \wedge dobx[4]$$

$$bxot1[2] = bx[2] \wedge dobx[6] \wedge dobx[4] \wedge dobx[3]$$

$$bxot1[1] = bx[1] \wedge dobx[5] \wedge dobx[3] \wedge dobx[2]$$

$$bxot1[0] = bx[0] \wedge dobx[4] \wedge dobx[2] \wedge dobx[1]$$

の演算で算出し、

また、前記演算結果データ $bxot1[7:0]$ 及び前記第 1 の演算ブロックの演算結果データ $gxot[7:0]$ に基づいて行った演算の結果を示す 8 ビット単位のバイトデータ $bxot2[7:0]$ を、

$$bxot2[7:0] = bxot1[7:0] \wedge gxot1d[7:0]$$

の演算で算出する

ことを特徴とする請求項 10 記載の誤り及び同期検出方法。

【請求項 13】 前記パリティチェック工程では、

前記データ並び替え工程から前記第 1 の演算工程に順次送られる 8 ビット単位のバイトデータを前記データ記憶装置に記憶すると共に特定の期間保持し、

前記第 1 の演算工程での演算結果の 8 ビット単位のバイトデータを前記データ記憶装置に記憶し、

更に、前記データ記憶装置から前記第 1 の演算工程での演算結果の 8 ビット単位のバイトデータを前記特定の期間保持した後、前記第 2 の演算工程に送る

ことを特徴とする請求項 10 記載の誤り及び同期検出方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、デジタルデータをフォーマット化する際の誤り検出及び同期検出を行なう装置並びに方法に関するものである。

【0002】

【従来の技術】

ケーブルモデムを用いたインターネット通信や北米を中心とするCATV放送におけるDown Stream伝送では、ITU-T J. 8 3 ANNEX B仕様に準拠したデータフォーマットが用いられている。この仕様では、7ビットを1バイトとするリードソロモン復号データから構成されるパケットデータを、8ビットを1バイトとするトランスポートストリームデータパケットに変換する処理を行なう。この際、7ビットデータパケットにおける同期バイト検出処理、誤り検出処理、及びトランスポートストリーム同期バイト挿入処理を行なうことが必要となる。これら7ビットを1バイトとするリードソロモン復号データから構成されるパケットデータを、8ビットを1バイトとするトランスポートストリームデータパケットに変換する一連の処理を以下ではMPEGフレーミング処理と呼ぶことにする。

【0003】

MPEGフレーミング処理は、ITU-T Recommendation J. 8 3 (ITU-T勧告書 J. 8 3)、ANNEX B、Digital multi-program System B、B.4 MPEG-2 transport framing の項に詳細に示されており、特に、FigureB.3/J. 8 3として、実際のデコード回路が提示されている。この回路は、同期バイト検出を行なうためのシンドローム計算回路である。

【0004】

この回路は、前記7ビットバイトのリードソロモン復号データを、一旦、シリアルビットに変換したデータを入力する仕様となっており、シリアルデータを用いてシンドローム計算を行なうことにより、誤り検出のためのパリティチェックを行なうと共に、このパリティチェックの結果に対応して同期バイト検出が同時に実行される回路となっている。そして、前記シリアル処理によるパリティチェック及び同期検出を行った後、出力データを8ビットを1バイトとする8ビット単位のバイトデータに変換して、トランスポートストリームデータパケットを構

成している。

【0005】

【発明が解決しようとする課題】

一般に、パリティチェック及び同期検出処理は、トランスポートストリームパケットにして、1パケット単位の処理となる。即ち、8ビット×188バイト＝1504ビット単位の処理となり、1パケットにおける最初の1バイトから最後の1バイトまでは、最小で1496ビット分の遅延が必要となる。

【0006】

このため、前記FigureB.3/J.83に提示されたデコード回路では、7ビット単位のバイトデータをシリアルビットデータに変換した後、1497段の遅延器を用いた処理を行い、その後、8ビット単位のバイトデータに変換するというシーケンスを用いている。従って、処理全体を実現するには、パラレル-シリアル変換回路、FigureB.3/J.83に提示された1497段の遅延器を用いたMPEG2-同期検出用シンδροーム計算回路、シリアル-パラレル変換回路が必要となり、回路規模が膨大となる欠点がある。

【0007】

本発明は、前記の問題点を解決するためになされたものであり、その目的は、1497段の遅延器、及び出力段のシリアル-パラレル変換回路を不要にできる誤り及び同期検出回路を提供することにある。

【0008】

【課題を解決するための手段】

前記目的を達成するために、本発明では、MPEGフレーミング処理全体としては7ビット単位のバイトデータに種々の処理を加えた後に最終的に8ビット単位のバイトデータにするというバイト-バイト変換である点に着目し、最初に7ビットバイトデータを8ビットバイトデータにバイト-バイト変換処理し、その後、8ビットのバイト単位でパリティチェック及び同期検出を行なうシーケンスを実現する。

【0009】

具体的に、請求項1記載の発明の誤り及び同期検出装置は、7ビット単位のバ

イトデータを入力して8ビット単位のバイトデータに変換するデータ並び替えブロックと、前記データ並び替えブロックで変換された8ビット単位のバイトデータを入力し、該バイトデータを用いてMPEG同期バイトの検出及びパリティチェックによる誤りの検出を行うパリティチェックブロックと、前記データ並び替えブロックで変換された8ビット単位のバイトデータ、及び前記パリティチェックブロックにおけるMPEG同期バイトの検出及びパリティチェックによる誤り検出のための演算途中の8ビット単位の中間バイトデータを入力して記憶し、8ビット単位のバイトデータを基本単位としてデータの入出力が可能なデータ記憶ブロックとを備えて、前記パリティチェックブロックから、同期バイトを含む8ビット単位のバイトデータの集合であるMPEGパケットデータを出力することを特徴とする。

【0010】

請求項2記載の発明は、前記請求項1記載の誤り及び同期検出装置において、前記データ並び替えブロックは、連続して入力される7ビット単位の2つのバイトデータを1組として、最初に入力されるバイトデータと後に入力されるバイトデータの上位1ビットとを組み合わせた8ビット単位の第1のバイトデータと、前記最初に入力されるバイトデータの下位6ビットと後に入力されるバイトデータの上位2ビットとを組み合わせた8ビット単位の第2のバイトデータと、同様に最初に入力されるバイトデータの下位 n ビット ($n=5, 4, 3, 2, 1$) と後に入力されるバイトデータの上位 m ビット ($m=3, 4, 5, 6, 7$) とを組み合わせた8ビット単位の第3～第7のバイトデータとを生成することを特徴とする。

【0011】

請求項3記載の発明は、前記請求項1記載の誤り及び同期検出装置において、前記パリティチェックブロックは、データの所定クロック分の遅延を含む所定のシンドローム計算を行う第1及び第2の演算ブロックを有し、前記第1の演算ブロックは、前記データ並び替えブロックから出力される8ビット単位のバイトデータを受けて、前記データの所定クロック分の遅延前の演算を行い、この演算結果である中間バイトデータを前記データ記憶ブロックに出力し、前記第2の演算

ブロックは、前記データ記憶ブロックから前記中間バイトデータを受けて、前記データの所定クロック分の遅延前の演算を行って、MPEG同期検出及びパリティチェックによる誤り検出を行った8ビット単位のバイトデータを出力することを特徴とする。

【0012】

請求項4記載の発明は、前記請求項3記載の誤り及び同期検出装置において、前記第1の演算ブロックは、入力される8ビット単位のバイトデータをpdatai[7:0]、このバイトデータpdatai[7:0]を用いて行った演算結果を示すデータをgxot[7:0]、この演算結果データgxot[7:0]を特定の基準クロックに従って7クロック分遅延させた8ビット単位のバイトデータをgxot7d[7:0]、前記演算結果データgxot[7:0]を計算する過程で使用する8ビット単位の間変数をgx[7:0]として、各ビットの排他的論理和を「^」で示すとき、該8ビット単位の間変数gx[7:0]を構成する各ビットgx[7]、gx[6]、gx[5]、gx[4]、gx[3]、gx[2]、gx[1]、gx[0]を、各々、

$$gx[0] = gxot7d[0]$$

$$gx[1] = gxot7d[1]$$

$$gx[2] = gxot7d[2] \wedge gxot7d[0]$$

$$gx[3] = gxot7d[3] \wedge gxot7d[1] \wedge gxot7d[0]$$

$$gx[4] = gxot7d[4] \wedge gxot7d[2] \wedge gxot7d[1]$$

$$gx[5] = gxot7d[5] \wedge gxot7d[3] \wedge gxot7d[2]$$

$$gx[6] = gxot7d[6] \wedge gxot7d[4] \wedge gxot7d[3]$$

$$gx[7] = gxot7d[7] \wedge gxot7d[5] \wedge gxot7d[4] \wedge gxot7d[0]$$

の演算により算出し、

前記演算結果データgxot[7:0]を構成する各ビットgxot[7]、gxot[6]、gxot[5]、gxot[4]、gxot[3]、gxot[2]、gxot[1]、gxot[0]を、前記中間変数gx[7:0]を用いて、各々、

$$gxot[7] = gx[7] \wedge pdatai[7]$$

$$gxot[6] = gx[7] \wedge gx[6] \wedge pdatai[7] \wedge pdatai[6]$$

$$gxot[5] = gx[7] \wedge gx[6] \wedge gx[5]$$

$$\begin{aligned}
 & \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \\
 \text{gxot}[4] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \\
 & \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \\
 \text{gxot}[3] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \\
 & \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \\
 \text{gxot}[2] &= \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \\
 & \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \\
 \text{gxot}[1] &= \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \\
 & \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \\
 \text{gxot}[0] &= \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \wedge \text{gx}[0] \\
 & \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \wedge \text{pdatai}[0]
 \end{aligned}$$

の演算により算出することを特徴とする。

【 0 0 1 3 】

請求項 5 記載の発明は、前記請求項 3 記載の誤り及び同期検出装置において、前記第 2 の演算ブロックは、入力される 8 ビット単位のバイトデータを $\text{dobx}[7:0]$ 、このバイトデータ $\text{dobx}[7:0]$ を用いて行った演算の結果を示すデータを $\text{bxot1}[7:0]$ 、前記入力されるバイトデータ $\text{dobx}[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $\text{dobx7d}[7:0]$ 、前記演算結果データ $\text{bxot1}[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $\text{bx}[7:0]$ 、前記第 1 の演算ブロックの演算結果データ $\text{gxot}[7:0]$ を前記基準クロックに従って 1 クロック分遅延させた 8 ビット単位のバイトデータを $\text{gxot1d}[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、前記 8 ビット単位の間変数 $\text{bx}[7:0]$ を構成する各ビット $\text{bx}[7]$ 、 $\text{bx}[6]$ 、 $\text{bx}[5]$ 、 $\text{bx}[4]$ 、 $\text{bx}[3]$ 、 $\text{bx}[2]$ 、 $\text{bx}[1]$ 、 $\text{bx}[0]$ を、各々、

$\text{bx}[0] = \text{dobx7d}[0]$

$\text{bx}[1] = \text{dobx7d}[1]$

$\text{bx}[2] = \text{dobx7d}[2]$

$\text{bx}[3] = \text{dobx7d}[3]$

$\text{bx}[4] = \text{dobx7d}[4]$

$$\begin{aligned} \text{bx}[5] &= \text{dobx7d}[5] \wedge \text{dobx}[1] \\ \text{bx}[6] &= \text{dobx7d}[6] \wedge \text{dobx}[2] \\ \text{bx}[7] &= \text{dobx7d}[7] \wedge \text{dobx}[3] \wedge \text{dobx}[1] \end{aligned}$$

の演算で算出し、

また、前記演算結果データ $\text{bxot1}[7:0]$ を構成する各ビット $\text{bxot1}[7]$ 、 $\text{bxot1}[6]$ 、 $\text{bxot1}[5]$ 、 $\text{bxot1}[4]$ 、 $\text{bxot1}[3]$ 、 $\text{bxot1}[2]$ 、 $\text{bxot1}[1]$ 、 $\text{bxot1}[0]$ を、各々、

$$\begin{aligned} \text{bxot1}[7] &= \text{bx}[7] \wedge \text{dobx}[0] \\ \text{bxot1}[6] &= \text{bx}[6] \wedge \text{bx}[0] \wedge \text{dobx}[7] \\ \text{bxot1}[5] &= \text{bx}[5] \wedge \text{dobx}[7] \wedge \text{dobx}[6] \\ \text{bxot1}[4] &= \text{bx}[4] \wedge \text{bx}[0] \wedge \text{dobx}[6] \wedge \text{dobx}[5] \\ \text{bxot1}[3] &= \text{bx}[3] \wedge \text{dobx}[7] \wedge \text{dobx}[5] \wedge \text{dobx}[4] \\ \text{bxot1}[2] &= \text{bx}[2] \wedge \text{dobx}[6] \wedge \text{dobx}[4] \wedge \text{dobx}[3] \\ \text{bxot1}[1] &= \text{bx}[1] \wedge \text{dobx}[5] \wedge \text{dobx}[3] \wedge \text{dobx}[2] \\ \text{bxot1}[0] &= \text{bx}[0] \wedge \text{dobx}[4] \wedge \text{dobx}[2] \wedge \text{dobx}[1] \end{aligned}$$

の演算で算出し、

また、前記演算結果データ $\text{bxot1}[7:0]$ 及び前記第 1 の演算ブロックの演算結果データ $\text{gxot}[7:0]$ に基づいて行った演算の結果を示す 8 ビット単位のバイトデータ $\text{bxot2}[7:0]$ を、

$$\text{bxot2}[7:0] = \text{bxot1}[7:0] \wedge \text{gxot1d}[7:0]$$

の演算で算出することを特徴とする。

【 0 0 1 4 】

請求項 6 記載の発明は、前記請求項 3 記載の誤り及び同期検出装置において、前記データ記憶ブロックは、前記データ並び替えブロックからの 8 ビット単位のバイトデータ、及び前記第 1 の演算ブロックの演算結果である 8 ビット単位のバイトデータを入力し、この両バイトデータを各々特定の期間保持した後に出力することを特徴とする。

【 0 0 1 5 】

請求項 7 記載の発明は、前記請求項 1 又は 6 記載の誤り及び同期検出装置において、前記データ記憶ブロックは RAM により構成されることを特徴とする。

【 0 0 1 6 】

請求項 8 記載の発明の誤り及び同期検出方法は、7ビット単位のバイトデータを入力して8ビット単位のバイトデータに変換するデータ並び替え工程と、前記データ並び替えブロックで変換された8ビット単位のバイトデータを入力し、該バイトデータを用いてシンドローム計算し、その計算途中の8ビット単位の中間バイトデータをデータ記憶装置に一時記憶させながら、該データ記憶装置に記憶した前記中間バイトデータを用いて前記シンドローム計算を続行して、MPEG同期バイトの検出及びパリティチェックによる誤りの検出を行うパリティチェック工程とを備え、同期バイトを含む8ビット単位のバイトデータの集合であるMPEGパケットデータを出力することを特徴とする。

【 0 0 1 7 】

請求項 9 記載の発明は、前記請求項 8 記載の誤り及び同期検出方法において、前記データ並び替え工程では、連続して入力される7ビット単位の2つのバイトデータを1組として、最初に入力されるバイトデータと後に入力されるバイトデータの上位1ビットとを組み合わせた8ビット単位の第1のバイトデータと、前記最初に入力されるバイトデータの下位6ビットと後に入力されるバイトデータの上位2ビットとを組み合わせた8ビット単位の第2のバイトデータと、同様に最初に入力されるバイトデータの下位 n ビット($n=5, 4, 3, 2, 1$)と後に入力されるバイトデータの上位 m ビット($m=3, 4, 5, 6, 7$)とを組み合わせた8ビット単位の第3～第7のバイトデータとを生成することを特徴とする。

【 0 0 1 8 】

請求項 10 記載の発明は、前記請求項 8 記載の誤り及び同期検出方法において、前記パリティチェック工程は、データの所定クロック分の遅延を含む所定のシンドローム計算を行う第1及び第2の演算工程を有し、前記第1の演算工程では、前記データ並び替え工程で変換された8ビット単位のバイトデータを受けて、前記データの所定クロック分の遅延前の演算を行い、この演算結果である8ビット単位の中間バイトデータを前記データ記憶装置に出力し、前記第2の演算工程では、前記データ記憶装置から前記中間バイトデータを受けて、前記データの所

定クロック分の遅延前の演算を行って、MPEG同期検出及びパリティチェックによる誤り検出を行った8ビット単位のバイトデータを出力することを特徴とする。

【0019】

請求項11記載の発明は、前記請求項10記載の誤り及び同期検出方法において、前記第1の演算工程では、入力される8ビット単位のバイトデータをpdatai[7:0]、このバイトデータpdatai[7:0]を用いて行った演算結果を示すデータをgxot[7:0]、この演算結果データgxot[7:0]を特定の基準クロックに従って7クロック分遅延させた8ビット単位のバイトデータをgxot7d[7:0]、前記演算結果データgxot[7:0]を計算する過程で使用する8ビット単位の間変数をgx[7:0]として、各ビットの排他的論理和を「^」で示すとき、該8ビット単位の間変数gx[7:0]を構成する各ビットgx[7]、gx[6]、gx[5]、gx[4]、gx[3]、gx[2]、gx[1]、gx[0]を、各々、

$$gx[0] = gxot7d[0]$$

$$gx[1] = gxot7d[1]$$

$$gx[2] = gxot7d[2] \wedge gxot7d[0]$$

$$gx[3] = gxot7d[3] \wedge gxot7d[1] \wedge gxot7d[0]$$

$$gx[4] = gxot7d[4] \wedge gxot7d[2] \wedge gxot7d[1]$$

$$gx[5] = gxot7d[5] \wedge gxot7d[3] \wedge gxot7d[2]$$

$$gx[6] = gxot7d[6] \wedge gxot7d[4] \wedge gxot7d[3]$$

$$gx[7] = gxot7d[7] \wedge gxot7d[5] \wedge gxot7d[4] \wedge gxot7d[0]$$

の演算により算出し、

前記演算結果データgxot[7:0]を構成する各ビットgxot[7]、gxot[6]、gxot[5]、gxot[4]、gxot[3]、gxot[2]、gxot[1]、gxot[0]を、前記中間変数gx[7:0]を用いて、各々、

$$gxot[7] = gx[7] \wedge pdatai[7]$$

$$gxot[6] = gx[7] \wedge gx[6] \wedge pdatai[7] \wedge pdatai[6]$$

$$gxot[5] = gx[7] \wedge gx[6] \wedge gx[5] \\ \wedge pdatai[7] \wedge pdatai[6] \wedge pdatai[5]$$

$$\begin{aligned}
 \text{gxot}[4] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \\
 &\quad \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \\
 \text{gxot}[3] &= \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \\
 &\quad \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \\
 \text{gxot}[2] &= \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \\
 &\quad \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \\
 \text{gxot}[1] &= \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \\
 &\quad \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \\
 \text{gxot}[0] &= \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \wedge \text{gx}[0] \\
 &\quad \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \wedge \text{pdatai}[0]
 \end{aligned}$$

の演算により算出することを特徴とする。

【 0 0 2 0 】

請求項 1 2 記載の発明は、前記請求項 1 0 記載の誤り及び同期検出方法において、前記第 2 の演算工程は、入力される 8 ビット単位のバイトデータを $\text{dobx}[7:0]$ 、このバイトデータ $\text{dobx}[7:0]$ を用いて行った演算の結果を示すデータを $\text{bxot1}[7:0]$ 、前記入力されるバイトデータ $\text{dobx}[7:0]$ を特定の基準クロックに従って 7 クロック分遅延させた 8 ビット単位のバイトデータを $\text{dobx7d}[7:0]$ 、前記演算結果データ $\text{bxot1}[7:0]$ を計算する過程で使用する 8 ビット単位の間変数を $\text{bx}[7:0]$ 、前記第 1 の演算ブロックの演算結果データ $\text{gxot1}[7:0]$ を前記基準クロックに従って 1 クロック分遅延させた 8 ビット単位のバイトデータを $\text{gxot1d}[7:0]$ として、各ビットの排他的論理和を「 \wedge 」で示すとき、前記 8 ビット単位の間変数 $\text{bx}[7:0]$ を構成する各ビット $\text{bx}[7]$ 、 $\text{bx}[6]$ 、 $\text{bx}[5]$ 、 $\text{bx}[4]$ 、 $\text{bx}[3]$ 、 $\text{bx}[2]$ 、 $\text{bx}[1]$ 、 $\text{bx}[0]$ を、各々、

$$\begin{aligned}
 \text{bx}[0] &= \text{dobx7d}[0] \\
 \text{bx}[1] &= \text{dobx7d}[1] \\
 \text{bx}[2] &= \text{dobx7d}[2] \\
 \text{bx}[3] &= \text{dobx7d}[3] \\
 \text{bx}[4] &= \text{dobx7d}[4] \\
 \text{bx}[5] &= \text{dobx7d}[5] \wedge \text{dobx}[1]
 \end{aligned}$$

$$bx[6] = dobx7d[6] \wedge dobx[2]$$

$$bx[7] = dobx7d[7] \wedge dobx[3] \wedge dobx[1]$$

の演算で算出し、

また、前記演算結果データ $bxot1[7:0]$ を構成する各ビット $bxot1[7]$ 、 $bxot1[6]$ 、 $bxot1[5]$ 、 $bxot1[4]$ 、 $bxot1[3]$ 、 $bxot1[2]$ 、 $bxot1[1]$ 、 $bxot1[0]$ を、各々、

$$bxot1[7] = bx[7] \wedge dobx[0]$$

$$bxot1[6] = bx[6] \wedge bx[0] \wedge dobx[7]$$

$$bxot1[5] = bx[5] \wedge dobx[7] \wedge dobx[6]$$

$$bxot1[4] = bx[4] \wedge bx[0] \wedge dobx[6] \wedge dobx[5]$$

$$bxot1[3] = bx[3] \wedge dobx[7] \wedge dobx[5] \wedge dobx[4]$$

$$bxot1[2] = bx[2] \wedge dobx[6] \wedge dobx[4] \wedge dobx[3]$$

$$bxot1[1] = bx[1] \wedge dobx[5] \wedge dobx[3] \wedge dobx[2]$$

$$bxot1[0] = bx[0] \wedge dobx[4] \wedge dobx[2] \wedge dobx[1]$$

の演算で算出し、

また、前記演算結果データ $bxot1[7:0]$ 及び前記第 1 の演算ブロックの演算結果データ $gxot[7:0]$ に基づいて行った演算の結果を示す 8 ビット単位のバイトデータ $bxot2[7:0]$ を、

$$bxot2[7:0] = bxot1[7:0] \wedge gxot1d[7:0]$$

の演算で算出することを特徴とする。

【 0 0 2 1 】

請求項 1 3 記載の発明は、前記請求項 1 0 記載の誤り及び同期検出方法において、前記パリティチェック工程では、前記データ並び替え工程から前記第 1 の演算工程に順次送られる 8 ビット単位のバイトデータを前記データ記憶装置に記憶すると共に、特定の期間保持し、前記第 1 の演算工程での演算結果の 8 ビット単位のバイトデータを前記データ記憶装置に記憶し、更に、前記データ記憶装置から前記第 1 の演算工程での演算結果の 8 ビット単位のバイトデータを前記特定の期間保持した後、前記第 2 の演算工程に送ることを特徴とする。

【 0 0 2 2 】

以上により、請求項 1 ないし請求項 1 3 記載の発明の誤り及び同期検出装置並

びに方法では、入力された 7 ビット単位の一連のバイトデータは先ずデータ並び替えブロックにより 8 ビット単位のバイトデータに並び替えられる。この 8 ビット単位の各バイトデータはデータ記憶ブロックに入力されて記憶されると共に、パリティチェックブロックにも入力される。前記データ記憶ブロックに入力された 8 ビット単位の各バイトデータは、所定期間（例えば 1 4 9 6 クロック）遅延されて前記パリティチェックブロックに入力される。このパリティチェックブロックは、8 ビット単位の各バイトデータ及びそれらの所定期間遅延されたバイトデータを用いて同期検出演算及びパリティチェック演算を行なう。

【0 0 2 3】

従って、バイトデータを 7 ビット単位から 8 ビット単位にバイト - バイト変換し、その後は一貫して 8 ビット単位のバイトデータを用いて同期検出演算及びパリティチェック演算が行われるので、従来のように出力段に備えるシリアルパラレル変換が不要となり、回路規模の削減が可能になる。

【0 0 2 4】

また、8 ビット単位のバイトデータを用いて処理するので、データ記憶ブロックとして RAM を用いることが可能であり、この RAM に記憶したバイトデータは所定期間遅延して出力すればよい。従って、従来のように 1 0 0 0 段以上の多段の遅延器が必要となる場合に比して、回路規模が極めて小規模になる。

【0 0 2 5】

更に、8 ビット構成のバイト単位で演算処理を行うので、CPU 処理等にも適した演算方法となり、特定のハードウェア構成に限定されないのは勿論のこと、ソフトウェアで実現することも可能である。

【0 0 2 6】

【発明の実施の形態】

以下、本発明の実施の形態の誤り及び同期検出装置について図面を参照しながら説明する。

【0 0 2 7】

図 1 は本実施の形態に係る誤り及び同期検出装置の構成を示すブロック図である。同図の誤り及び同期検出装置は、データ並び替えブロック 1 と、パリティチ

ェックブロック 2 と、データ記憶ブロック（データ記憶装置） 3 から構成されている。前記データ並び替えブロック 1 は、入力された 7 ビット単位のバイトデータを 8 ビット単位のバイトデータに変換する。また、前記パリティチェックブロック 2 は、連続して入力される 8 ビット単位のバイトデータを用いて演算することにより、MPEG 同期バイト検出及びパリティチェックによる誤り検出を行なう。更に、データ記憶ブロック 3 は、パリティチェックブロック 2 における MPEG 同期バイト検出演算及びパリティチェックによる誤り検出演算を行なう際に、データの保持が必要であるため、このデータの保持を行なう。

【 0 0 2 8 】

図 2 は前記データ並び替えブロック 1 の内部構成を示し、図 3 はこのデータ並び替えブロック 2 のデータ変換方法を図示したものである。図 4 は前記パリティチェックブロック 2 の内部構成を示すブロック図である。

【 0 0 2 9 】

前記パリティチェックブロック 2 は、第 1 の演算ブロック 1 0 と、第 2 の演算ブロック 1 1 と、選択回路 1 2 とから構成される。前記第 1 及び第 2 の演算ブロック 1 0 の機能を概述すると、第 1 の演算ブロック 1 0 は、データ並び替えブロック 1 から 8 ビット単位のバイトデータを受けて演算を行って、データ記憶ブロック 3 に演算結果のバイトデータを出力する。第 2 の演算ブロック 1 1 は、データ記憶ブロック 3 から前記第 1 の演算ブロック 1 0 の演算結果の 8 ビット単位のバイトデータを受けて所定時間経過後に受けて演算を行い、MPEG 同期バイトを含んだデータを出力する。選択回路 1 2 は、第 2 の演算ブロック 1 1 の演算結果データ及びデータ記憶ブロック 3 から出力されるバイトデータの何れか一方を選択して、8 ビットの MPEG 2 トランスポートストリームデータとして出力する。

【 0 0 3 0 】

以下、処理の順番に従ってデータ並び替えブロック 1、パリティチェックブロック 2 及びデータ記憶ブロック 3 の構成及び動作を具体的に説明する。

【 0 0 3 1 】

先ず、7 ビットバイトの入力データがデータ並び替えブロック 1 に入力される

。この7ビット単位のバイトデータを1ビットずつシフトさせ、8ビットのデータを作成する。このバイトデータを生成するパターンとしては、合計8種類が生成される。図3に入力データ変換方法を示した過程図を示す。先ず、7ビット単位のデータをシリアル化し、1クロック毎に1ビットシフトさせた8ビットデータ（A、B…、J…）を形成する。図3から判るように、8クロック毎に同種類のデータが形成される。つまりAから始まる8ビット単位のバイトデータとIから始まる8ビット単位のバイトデータとは同じ組合せパターンであり、同様にBから始まる8ビット単位のバイトデータとJから始まる8ビット単位のバイトデータとは同じ組合せパターンのデータである。

【 0 0 3 2 】

前記の動作を行うデータ並び替えブロック1の内部構成を図2に示す。同図のブロック1は、シリアル化された7ビット単位のデータを1単位として格納する2個のレジスタ1a、1bと、セクタ1cとを有する。レジスタ1a、1bは直列に接続される。従って、レジスタ1bに7ビット単位の1バイトデータ（前バイトデータ）が格納されると、レジスタ1aにはそれに続く7ビット単位の1バイトデータ（後バイトデータ）が格納される。前記セクタ1cは第1～第7の入力端子を持ち、各入力端子を順番に選択する。第1の入力端子には前バイトデータの全ビットdataireg2[6:0] と後バイトデータの上位1ビットdataireg1[6] との合計8ビットが入力される。第2の入力端子には前バイトデータの下位6ビットdataireg2[5:0] と後バイトデータの上位2ビットdataireg1[6:5] との合計8ビットが、第3の入力端子には前バイトデータの下位5ビットdataireg2[4:0] と後バイトデータの上位3ビットdataireg1[6:4] との合計8ビットが、第4の入力端子には前バイトデータの下位4ビットdataireg2[3:0] と後バイトデータの上位4ビットdataireg1[6:3] との合計8ビットが、第5の入力端子には前バイトデータの下位3ビットdataireg2[2:0] と後バイトデータの上位5ビットdataireg1[6:2] との合計8ビットが、第6の入力端子には前バイトデータの下位2ビットdataireg2[1:0] と後バイトデータの上位6ビットdataireg1[6:1] との合計8ビットが、第7の入力端子には前バイトデータの最下位ビットdataireg2[0] と後バイトデータの全ビットdataireg1[6:0] との合計8ビットが各々入力さ

れる。

【0033】

このようにして、バイト並び替えブロック1において、7ビット単位のバイトデータから8ビット単位のバイトデータに並び替えられたデータがパリティチェックブロック2に出力される。

【0034】

パリティチェックブロック2は、図4に示すように、第1の演算ブロック10と第2の演算ブロック11と選択回路12とから構成される。前記第1の演算ブロック10及び第2の演算ブロック11は、ITU-T勧告書 J.83 ANNEX B、Digital multi-program System B、B.4 MPEG-2 transport framing の項に示された具体的なデコード回路と同等の演算を行う。この具体的なデコード回路の構成を図13に示す。同図のデコード回路は、シンδροームを次の方程式に基づいて計算する回路である。

【0035】

$$f(x) = [1 + x^{1497} b(x)] / g(x)$$

$$\text{ここに、} g(x) = 1 + x + x^5 + x^6 + x^8$$

$$b(x) = 1 + x + x^3 + x^7$$

である。

【0036】

データ並び替えブロック1から入力される8ビット単位のバイトデータは、先ず、第1の演算ブロック10に入力されると同時に、データ記憶ブロック3にも入力されて保持される。

【0037】

前記第1及び第2の演算ブロック10、11は前記方程式に基づいたシンδροーム計算を行うに際して、データ並び替えブロック1から入力される8ビット単位のバイトデータを単位としてパラレル処理を行う。第1の演算ブロック10は、図13のデコード回路のうち1497段の遅延器の前段(入力側)の回路部と同等の処理を行い、第2の演算ブロック11は、図13のデコード回路のうち1497段の遅延器の後段(出力側)の回路部と同等の処理を行う。この処理を行うため

に、先ず、図 1 3 のデコード回路のうち 1 4 9 7 段の遅延器の前段(入力側)の回路部(この回路図を図 9 に示す)の処理を解析する。

【 0 0 3 8 】

図 9 に示した回路部において、Point0～Point8における値の経時変化を図 1 0 に示す。同図は、図 9 においてある時点での各Point7～Point0の値をX7～X0とし、またその時点からシリアル入力データa7、a6、…a0が入力されるとし、特定の基準クロックに従って8クロック分状態が変化したときの様子を示している。時間はtime1、time2、time3 … の順に変化するとする。シリアルデータの演算処理結果はPoint 8 の値である。このPoint 8 におけるtime1～time 8 のシリアル処理演算を前記第 1 の演算ブロック 1 0 では、ある時刻において、同時に平行に行うことにより、平行演算を実現する。

【 0 0 3 9 】

即ち、第 1 の演算ブロック 1 0 に入力された 8 ビット単位のバイトデータは、以下の計算式によってバイト処理される。具体的には、入力される 8 ビット単位のバイトデータをpdatai[7:0]、このバイトデータpdatai[7:0]を用いて行った演算結果を示すデータをgxot[7:0]、この演算結果データgxot[7:0]を特定の基準クロックに従って7クロック分遅延させた8ビット単位のバイトデータをgxot7d[7:0]、前記演算結果データgxot[7:0]を計算する過程で使用する8ビット単位の間変数をgx[7:0]として、各ビットの排他的論理和を「^」で示すとき、該8ビット単位の間変数gx[7:0]を構成する各ビットgx[7]、gx[6]、gx[5]、gx[4]、gx[3]、gx[2]、gx[1]、gx[0]は、各々、

$$gx[0] = gxot7d[0] \quad \dots (1-0)$$

$$gx[1] = gxot7d[1] \quad \dots (1-1)$$

$$gx[2] = gxot7d[2] \wedge gxot7d[0] \quad \dots (1-2)$$

$$gx[3] = gxot7d[3] \wedge gxot7d[1] \wedge gxot7d[0] \quad \dots (1-3)$$

$$gx[4] = gxot7d[4] \wedge gxot7d[2] \wedge gxot7d[1] \quad \dots (1-4)$$

$$gx[5] = gxot7d[5] \wedge gxot7d[3] \wedge gxot7d[2] \quad \dots (1-5)$$

$$gx[6] = gxot7d[6] \wedge gxot7d[4] \wedge gxot7d[3] \quad \dots (1-6)$$

$$gx[7] = gxot7d[7] \wedge gxot7d[5] \wedge gxot7d[4] \wedge gxot7d[0]$$

... (1-7)

の演算により算出し、

また、前記gxot[7:0]を構成する各ビットgxot[7]、gxot[6]、gxot[5]、gxot[4]、gxot[3]、gxot[2]、gxot[1]、gxot[0]は、前記中間変数gx[7:0]を用いて、各々、

$$\text{gxot}[7] = \text{gx}[7] \wedge \text{pdatai}[7] \quad \dots (2-7)$$

$$\text{gxot}[6] = \text{gx}[7] \wedge \text{gx}[6] \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \quad \dots (2-6)$$

$$\begin{aligned} \text{gxot}[5] = & \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \\ & \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \quad \dots (2-5) \end{aligned}$$

$$\begin{aligned} \text{gxot}[4] = & \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \\ & \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \\ & \dots (2-4) \end{aligned}$$

$$\begin{aligned} \text{gxot}[3] = & \text{gx}[7] \wedge \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \\ & \wedge \text{pdatai}[7] \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \\ & \dots (2-3) \end{aligned}$$

$$\begin{aligned} \text{gxot}[2] = & \text{gx}[6] \wedge \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \\ & \wedge \text{pdatai}[6] \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \\ & \dots (2-2) \end{aligned}$$

$$\begin{aligned} \text{gxot}[1] = & \text{gx}[5] \wedge \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \\ & \wedge \text{pdatai}[5] \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \\ & \dots (2-1) \end{aligned}$$

$$\begin{aligned} \text{gxot}[0] = & \text{gx}[4] \wedge \text{gx}[3] \wedge \text{gx}[2] \wedge \text{gx}[1] \wedge \text{gx}[0] \\ & \wedge \text{pdatai}[4] \wedge \text{pdatai}[3] \wedge \text{pdatai}[2] \wedge \text{pdatai}[1] \wedge \text{pdatai}[0] \\ & \dots (2-0) \end{aligned}$$

の各演算により演算される。

【 0 0 4 0 】

図5は、前記第1の演算ブロック10の内部構成を示す。同図の排他的論理和回路（以下、XOR回路と称す）10aは前記演算式（2-7）を演算する。また、XOR回路10bは演算式（2-6）を、XOR回路10cは演算式（2-5）を、

XOR回路10dは演算式(2-4)を、XOR回路10eは演算式(2-3)を、XOR回路10fは演算式(2-2)を、XOR回路10gは演算式(2-1)を、XOR回路10hは演算式(2-0)を各々演算する。また、XOR回路10iは演算式(1-7)を、XOR回路10jは演算式(1-6)を、XOR回路10kは演算式(1-5)を、XOR回路10lは演算式(1-4)を、XOR回路10mは演算式(1-3)を、XOR回路10nは演算式(1-2)を、XOR回路10oは演算式(1-1)を、XOR回路10pは演算式(1-0)を各々演算する。更に、遅延回路10qは演算結果データgxot[7]を7クロック遅延し、遅延回路10rは演算結果データgxot[6]を7クロック遅延し、遅延回路10sは演算結果データgxot[5]を7クロック遅延し、遅延回路10tは演算結果データgxot[4]を7クロック遅延し、遅延回路10uは演算結果データgxot[3]を7クロック遅延し、遅延回路10vは演算結果データgxot[2]を7クロック遅延し、遅延回路10wは演算結果データgxot[1]を7クロック遅延し、遅延回路10xは演算結果データgxot[0]を7クロック遅延する。

【0041】

図10のPoint8におけるtime1～time8の値は、第1の演算ブロック10におけるビット7～ビット0と一致している。

【0042】

前記第1の演算ブロック10で処理された8ビット単位のバイトデータは、データ記憶ブロック3に送られ、1496クロック分遅延された後、第2の演算ブロック11に送られる。

【0043】

次に、第2の演算ブロック11の処理を説明する。この演算ブロック11の処理は、既述の通り図13に示したデコード回路のうち1497段の遅延器の後段(出力側)の回路部と同等の処理を行う。図11は、図13のデコード回路のうち1497段の遅延器の後段(出力側)の回路部を示し、以下、図11の回路部の処理を解析する。図11に示したPoint0～Point9における値の経時変化を図12に示す。同図では、図11においてある時点での各Point7～Point0の値をX7～X0とし、また、その時点からシリアル入力データa7、a6、…a0が入力されるとし、特

定の基準クロックに従って、8クロック分状態が変化したときの様子を示している。time1、time2、time3 …の順に時間が変化する。シリアルデータの演算処理結果はPoint8の値である。このPoint8におけるtime1～time8のシリアル処理演算を、前記第2の演算ブロック11では、ある時刻において、同時に平行に行う平行演算を行う。また、Point9における演算は、第1の演算ブロック10の出力結果データを1クロック分遅延させたデータと、Point8の演算結果データとを用いて、time1～time8のシリアル処理を行うものである。このPoint8及びPoint9におけるtime1～time8のシリアル処理演算を、前記第2の演算ブロック11ではある時刻において同時に平行に行う平行演算を実現する。

【0044】

即ち、第2の演算ブロック11に入力された8ビット単位のバイトデータは、以下の計算式によってバイト処理される。入力される8ビット単位のバイトデータをdobx[7:0]、このバイトデータdobx[7:0]を用いて行った演算の結果を示すデータをbxot1[7:0]、前記入力されるバイトデータdobx[7:0]を特定の基準クロックに従って7クロック分遅延させた8ビット単位のバイトデータをdobx7d[7:0]、前記演算結果データbxot1[7:0]を計算する過程で使用する8ビット単位の中間変数をbx[7:0]、前記第1の演算ブロック10の演算結果データgxot[7:0]を前記基準クロックに従って1クロック分遅延させた8ビット単位のバイトデータをgxot1d[7:0]として、各ビットの排他的論理和を「^」で示すとき、前記8ビット単位の間変数bx[7:0]を構成する各ビットbx[7]、bx[6]、bx[5]、bx[4]、bx[3]、bx[2]、bx[1]、bx[0]は、各々、

$$bx[0] = dobx7d[0] \quad \dots (3-0)$$

$$bx[1] = dobx7d[1] \quad \dots (3-1)$$

$$bx[2] = dobx7d[2] \quad \dots (3-2)$$

$$bx[3] = dobx7d[3] \quad \dots (3-3)$$

$$bx[4] = dobx7d[4] \quad \dots (3-4)$$

$$bx[5] = dobx7d[5] \wedge dobx[1] \quad \dots (3-5)$$

$$bx[6] = dobx7d[6] \wedge dobx[2] \quad \dots (3-6)$$

$$bx[7] = dobx7d[7] \wedge dobx[3] \wedge dobx[1] \quad \dots (3-7)$$

の演算で算出し、

また、前記**bxot1**[7:0]を構成する各ビット**bxot1**[7]、**bxot1**[6]、**bxot1**[5]、**bxot1**[4]、**bxot1**[3]、**bxot1**[2]、**bxot1**[1]、**bxot1**[0]は、前記中間変数**bx**[7:0]及び8ビットバイト入力データ**dobx**[7:0]を用いて、各々、

$$\text{bxot1}[7] = \text{bx}[7] \wedge \text{dobx}[0] \quad \dots (4-7)$$

$$\text{bxot1}[6] = \text{bx}[6] \wedge \text{bx}[0] \wedge \text{dobx}[7] \quad \dots (4-6)$$

$$\text{bxot1}[5] = \text{bx}[5] \wedge \text{dobx}[7] \wedge \text{dobx}[6] \quad \dots (4-5)$$

$$\text{bxot1}[4] = \text{bx}[4] \wedge \text{bx}[0] \wedge \text{dobx}[6] \wedge \text{dobx}[5] \quad \dots (4-4)$$

$$\text{bxot1}[3] = \text{bx}[3] \wedge \text{dobx}[7] \wedge \text{dobx}[5] \wedge \text{dobx}[4] \quad \dots (4-3)$$

$$\text{bxot1}[2] = \text{bx}[2] \wedge \text{dobx}[6] \wedge \text{dobx}[4] \wedge \text{dobx}[3] \quad \dots (4-2)$$

$$\text{bxot1}[1] = \text{bx}[1] \wedge \text{dobx}[5] \wedge \text{dobx}[3] \wedge \text{dobx}[2] \quad \dots (4-1)$$

$$\text{bxot1}[0] = \text{bx}[0] \wedge \text{dobx}[4] \wedge \text{dobx}[2] \wedge \text{dobx}[1] \quad \dots (4-0)$$

の演算で算出する。

【 0 0 4 5 】

更に、**bxot2**[7:0]は、該**bxot1**[7:0]及び前記**gxot1d**[7:0]を用いて、

$$\text{bxot2}[7:0] = \text{bxot1}[7:0] \wedge \text{gxot1d}[7:0] \quad \dots (5)$$

の演算で算出する。

【 0 0 4 6 】

前記の演算を行う第2の演算ブロック11の内部構成を図6に示す。同図において、XOR回路11aは前記演算式(3-7)を演算する。また、XOR回路11bは演算式(3-6)を、XOR回路11cは演算式(3-5)を、XOR回路11dは演算式(4-7)を、XOR回路11eは演算式(4-6)を、XOR回路11fは演算式(4-5)を、XOR回路11gは演算式(4-4)を、XOR回路11hは演算式(4-3)を、XOR回路11iは演算式(4-2)を、XOR回路11jは演算式(4-1)を、XOR回路11kは演算式(4-0)を各々演算する。更に、XOR回路11lは演算式(5)を演算する。加えて、遅延回路11mは入力された8ビット単位のバイトデータ**dobx**[7:0]を7クロック分遅延し、遅延回路11nは前記第1の演算回路10の演算結果データ**gxot**[7:0]を1クロック分遅延する遅延回路である。

【 0 0 4 7 】

図 1 2 の Point8 における time1 ~ time8 の値は、第 2 の演算ブロック 1 1 の出力結果データにおけるビット 7 ~ ビット 0 と一致している。

【 0 0 4 8 】

本実施の形態におけるパリティチェックブロック 2 は、第 1 の演算ブロック 1 0 及び第 2 の演算ブロック 1 0 でバイトデータの演算を行なうことにより、パリティチェックを行い、8 ビット単位のバイトデータが 1 8 8 バイト送信される毎に誤りが検出されなければ、47hex を出力し、誤りが検出された場合には 47hex を出力しない。つまり、結果として 47hex を検出して、同期バイト検出処理を行うことにより、パリティチェックを行なうことが可能となる。

【 0 0 4 9 】

パリティチェックブロック 2 中の選択回路 1 2 は、データ記憶ブロック 3 から入力される 8 ビット単位のバイトデータと、第 2 の演算ブロック 1 1 から出力されるパリティチェック結果を示すデータとを受けて、第 2 の演算ブロック 1 1 の出力結果データがトランスポートストリームパケットの先頭の同期バイトを示す 47hex のときには、この 47hex データを選択し、トランスポートストリームパケットの先頭でないデータの場合には、データ記憶ブロック 3 から入力される通常の 8 ビット単位のバイトデータを選択して出力する。

【 0 0 5 0 】

次に、データ記憶ブロック 3 の構成及び動作について説明する。本実施の形態では、データ記憶ブロック 3 は、1 4 9 6 ワード、1 6 ビットの 2port-RAM が用いられ、入力信号と計算データとを 1 4 9 6 クロック分遅延させるために用いる。

【 0 0 5 1 】

データ記憶ブロック 3 の構成を図 7 (a) に、データ記憶ブロック 3 内のデータの経時変化を同図 (b) に、入出力タイミングチャートを図 8 に各々示す。データ記憶ブロック (RAM) 3 には、図 7 (a) に示すように、データ di [15:0] (1 6 ビットバイト) が入力される。この入力データ di [15:0] は、データ並び替えブロック 1 の出力データ pdatai [7:0] (8 ビットバイト) を下位ビットとし、パリ

ティチェックブロック 2 中の第 1 の演算ブロック 1 0 の出力データ $gxot[7:0]$ (8 ビットバイト) を上位ビットとしてビット連結して成る。また、同 RAM 3 には、Read イネーブル信号 nre 、Write イネーブル信号 nwe 、Read 用アドレス $addrb[10:0]$ 、及び Write 用アドレス $addra[10:0]$ が入力される。図 7 (b) から判るように、Write 用アドレス $addra[10:0]$ は、Read 用アドレス $addrb[10:0]$ を 1 クロック遅延させており、常に同一アドレスに対して Read した後に Write するように構成される。Read イネーブル信号 nre 及び Write イネーブル信号 nwe は、非同期中は常に Low (Low-active) であり、同期中は正しいデータが来た時にのみ、つまり 8 クロックに 1 回 Low になる。RAM 3 の入力データ $di[15:0]$ は、入力時から 1 4 9 6 クロック分遅延した時点で出力データ $do[15:0]$ として出力される。この出力データ $do[15:0]$ は、1 クロック分遅れて (1 ラッチして) データ $doq[15:0]$ としてパリティチェックブロック 2 に取り込まれ、前記出力データ $do[15:0]$ の上位 8 ビットのデータ、即ち第 1 の演算ブロック 1 0 の出力データ $gxot[7:0]$ が第 2 の演算ブロック 1 1 の演算に供される。

【 0 0 5 2 】

以上説明したように、本実施の形態によれば、7 ビット単位のバイトデータをデータ並び替えブロック 1 で 8 ビット単位のバイトデータに変換した後、パリティチェックブロック 2 及びデータ記憶ブロック 3 に入力し、先ず、パリティチェックブロック 2 では、8 ビット構成のバイト単位での演算により同期検出を行なうことが可能となると同時に、8 ビット構成のバイト単位での演算によりパリティチェックを行なうことが可能となる。更に、データ記憶ブロック 3 についても、8 ビット構成のバイト単位でデータを取り扱うことが可能となり、データの演算処理からトランスポートストリーム出力までを一貫して 8 ビット構成のバイト単位で取り扱うことが可能となる。

【 0 0 5 3 】

また、本発明における 8 ビット構成のバイト単位での計算処理方法は、ハードウェアとして CPU を用いた場合に、ソフトウェアとして実現することも可能であり、ソフトウェアにより 8 ビット構成のバイト単位での処理を可能とする誤り及び同期検出方法として実現することが可能である。

【 0 0 5 4 】

【発明の効果】

以上説明したように、請求項 1 ないし請求項 1 3 記載の誤り及び同期検出装置並びに方法によれば、バイトデータを 7 ビット単位から 8 ビット単位にバイト - バイト変換し、その後は一貫して 8 ビット単位のバイトデータを用いて同期検出演算及びパリティチェック演算を行ったので、従来のように出力段に備えるシリアルパラレル変換が不要となり、回路規模を効果的に削減できる。

【 0 0 5 5 】

しかも、8 ビット単位のバイトデータを用いて処理するので、データ記憶ブロックとして RAM を用いることが可能であって、多段の遅延器に代えて RAM に記憶したバイトデータを所定期間遅延して出力すればよく、従来の多段の遅延器が不要となり、回路規模を極めて小規模にできる。

【 0 0 5 6 】

更に、8 ビット構成のバイト単位で演算処理を行うので、CPU 処理等にも適した演算方法となり、種々のハードウェア構成で実現できると共に、ソフトウェアで実現することも可能である。

【図面の簡単な説明】

【図 1】

本発明の実施の形態の誤り及び同期検出装置の概略構成を示すブロック図である。

【図 2】

同誤り及び同期検出装置に備えるデータ並び替えブロックの構成を示す図である。

【図 3】

同データ並び替えブロックによるデータ並び替え方法を示す図である。

【図 4】

同誤り及び同期検出装置に備えるパリティチェックブロックの構成を示すブロック図である。

【図 5】

同パリティチェックブロックに備える第1の演算ブロックの構成を示す図である。

【図6】

同パリティチェックブロックに備える第2の演算ブロックの構成を示す図である。

【図7】

(a) は同誤り及び同期検出装置に備えるデータ記憶ブロックを構成するのRAMのデータ記憶状況を示す図、(b) は同データ記憶ブロックのデータの入出力の経時変化を示す図である。

【図8】

同データ記憶ブロックの動作タイミングチャートを示す図である。

【図9】

MPEGフレーミング処理についてITU-T勧告書 J. 8 3 が提案するデコード回路の1497段の遅延前の回路部分を示す図である。

【図10】

同回路部分による演算過程を示す図である。

【図11】

MPEGフレーミング処理についてITU-T勧告書 J. 8 3 が提案するデコード回路の1497段の遅延後の回路部分を示す図である。

【図12】

同回路部分による演算過程を示す図である。

【図13】

MPEGフレーミング処理についてITU-T勧告書 J. 8 3 が提案するデコード回路の全体構成を示す図である。

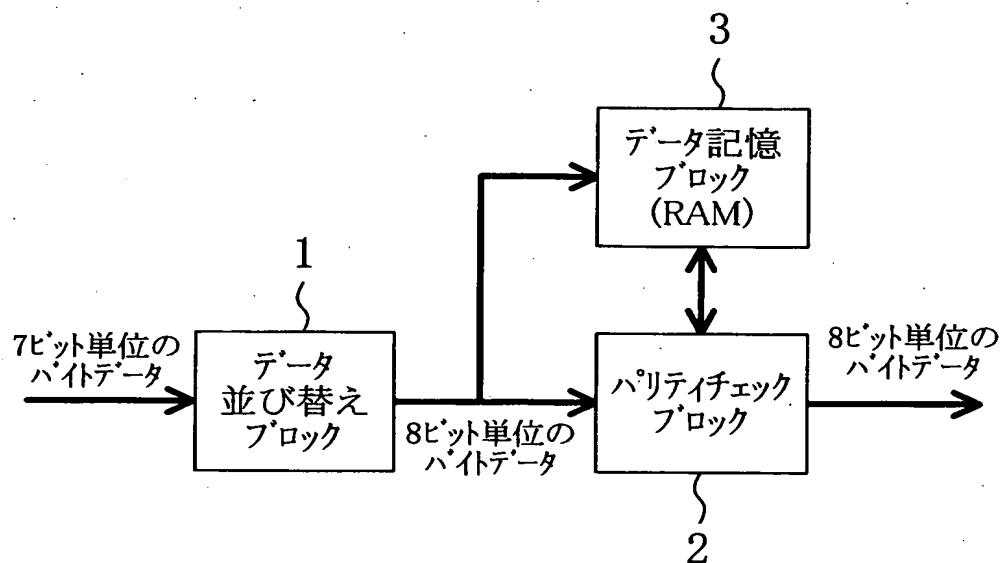
【符号の説明】

1	データ並び替えブロック
1 a、1 b	レジスタ
1 c	セレクタ
2	パリティチェックブロック

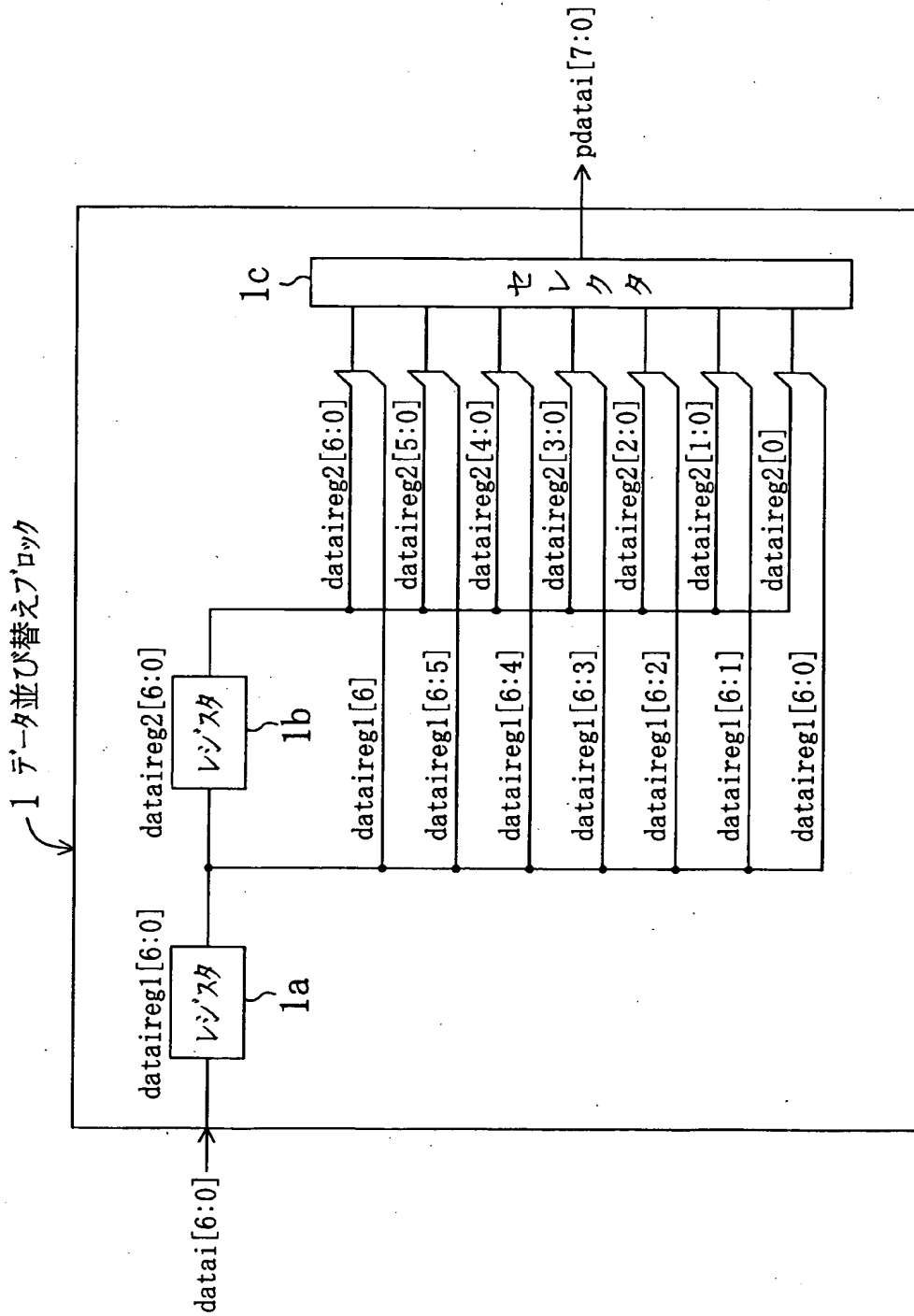
3	データ記憶ブロック (データ記憶装置)
1 0	第 1 の演算ブロック
1 0 a ~ 1 0 p	X O R 回路
1 0 q ~ 1 0 x	遅延回路
1 1	第 2 の演算ブロック
1 1 a ~ 1 1 k	X O R 回路
1 1 l、1 1 m	遅延回路
1 2	選択回路

【書類名】 図面

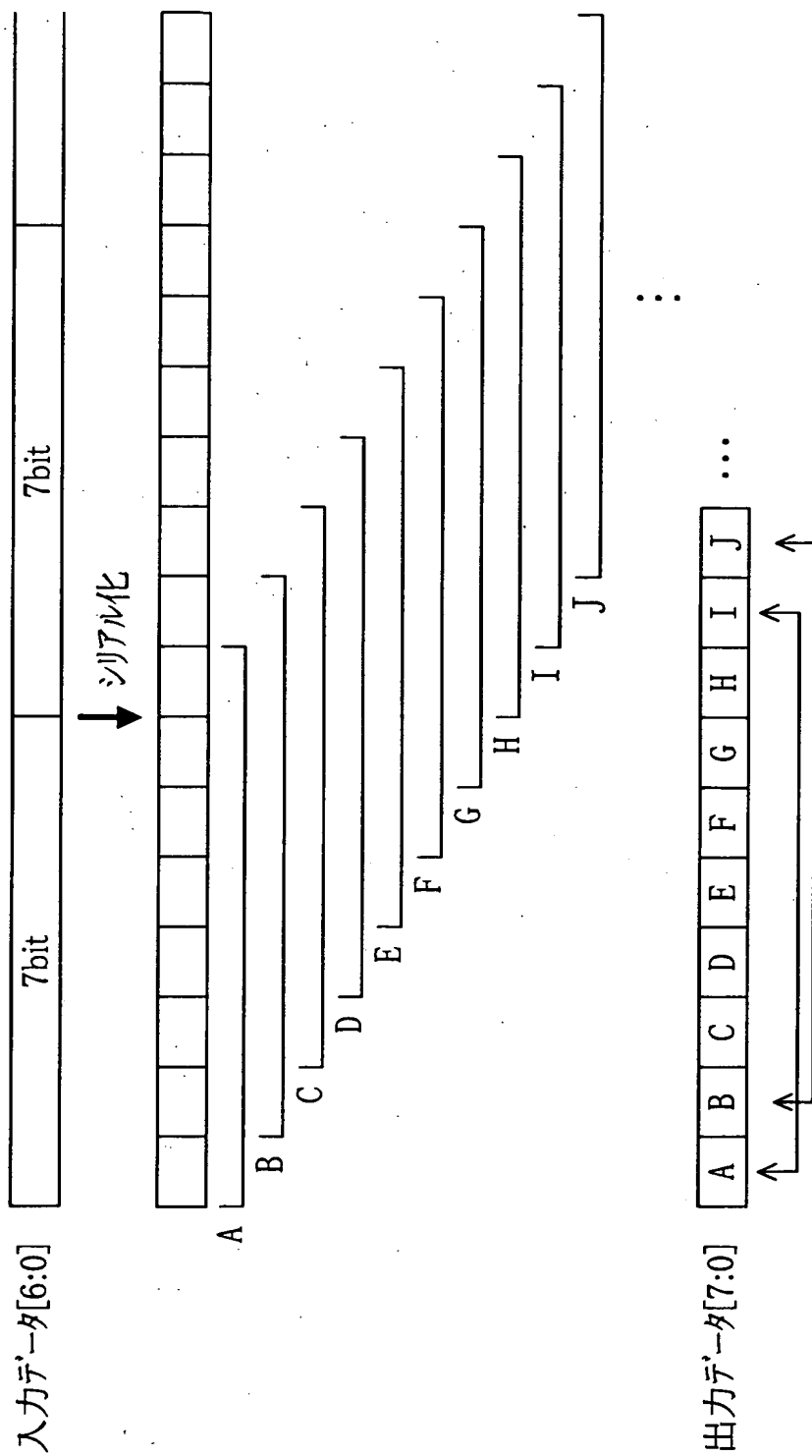
【図 1】



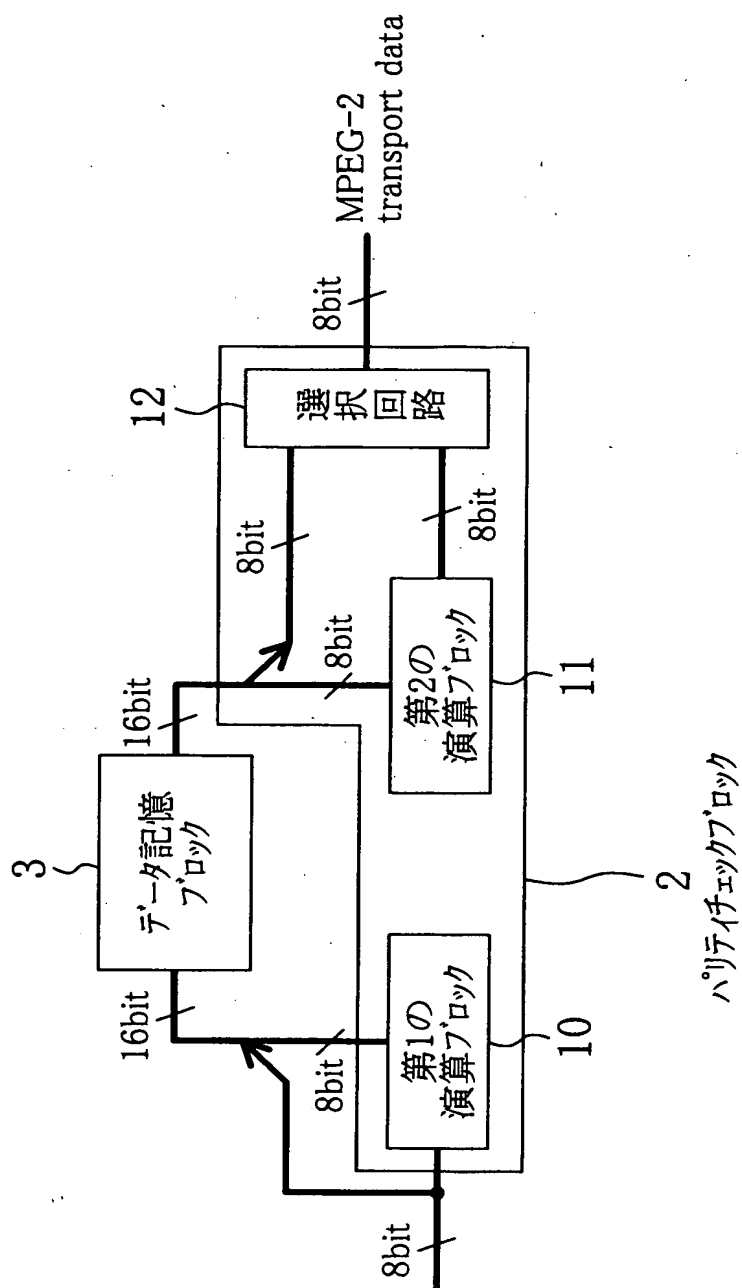
【図 2】



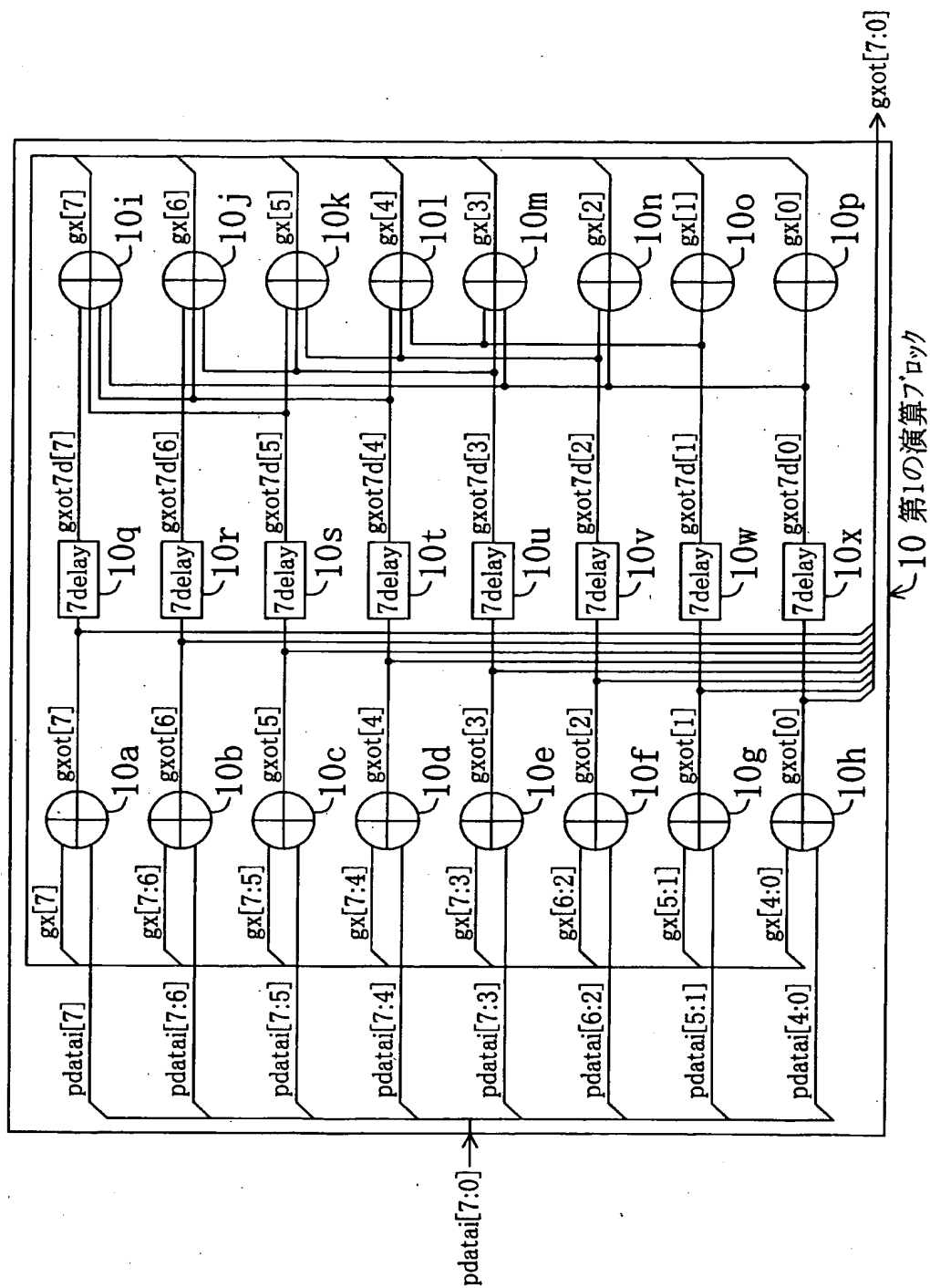
【図 3】



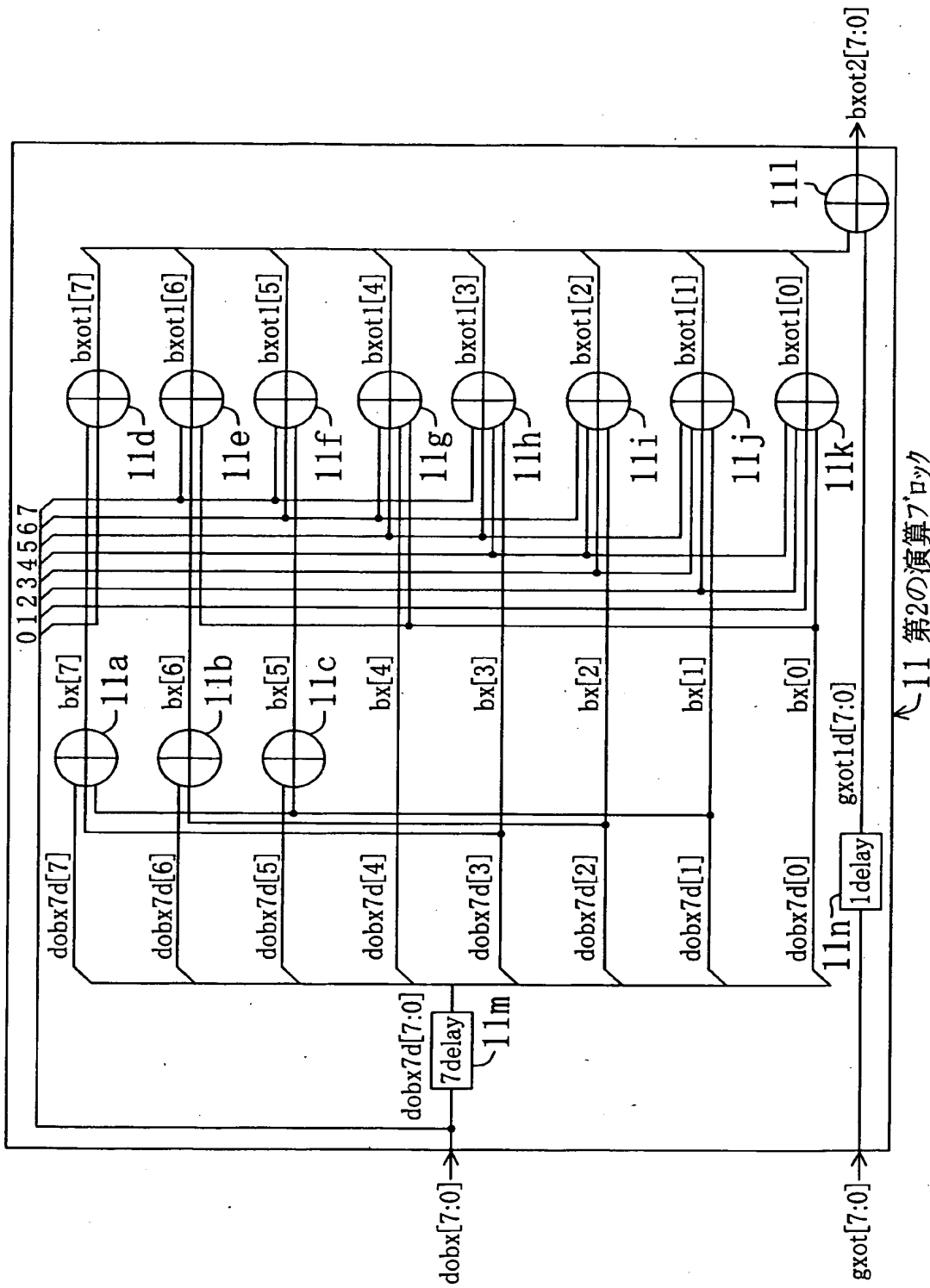
【図4】



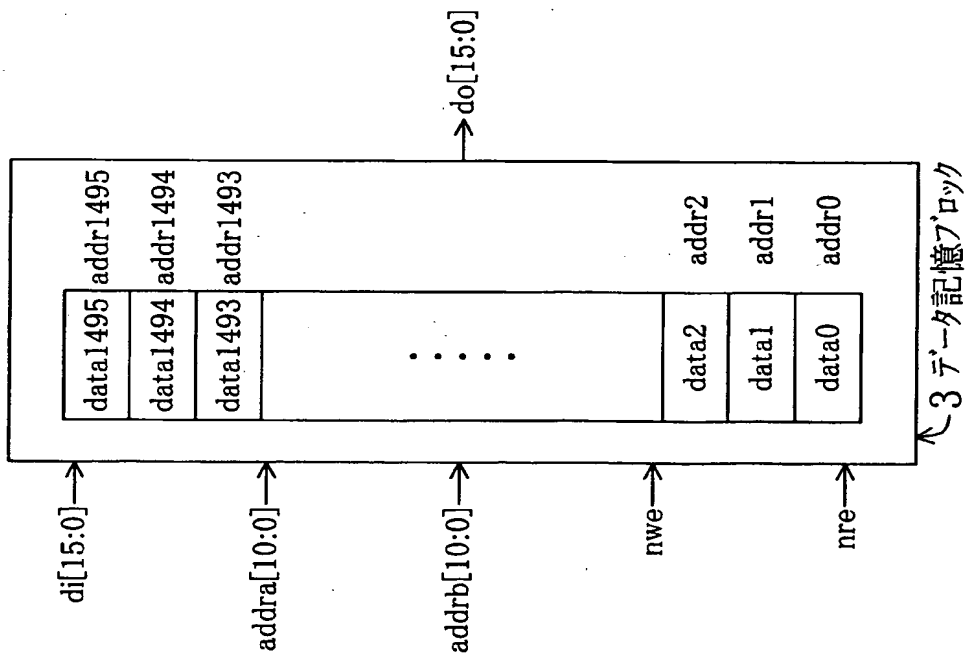
【図 5】



【図 6】



【図 7】

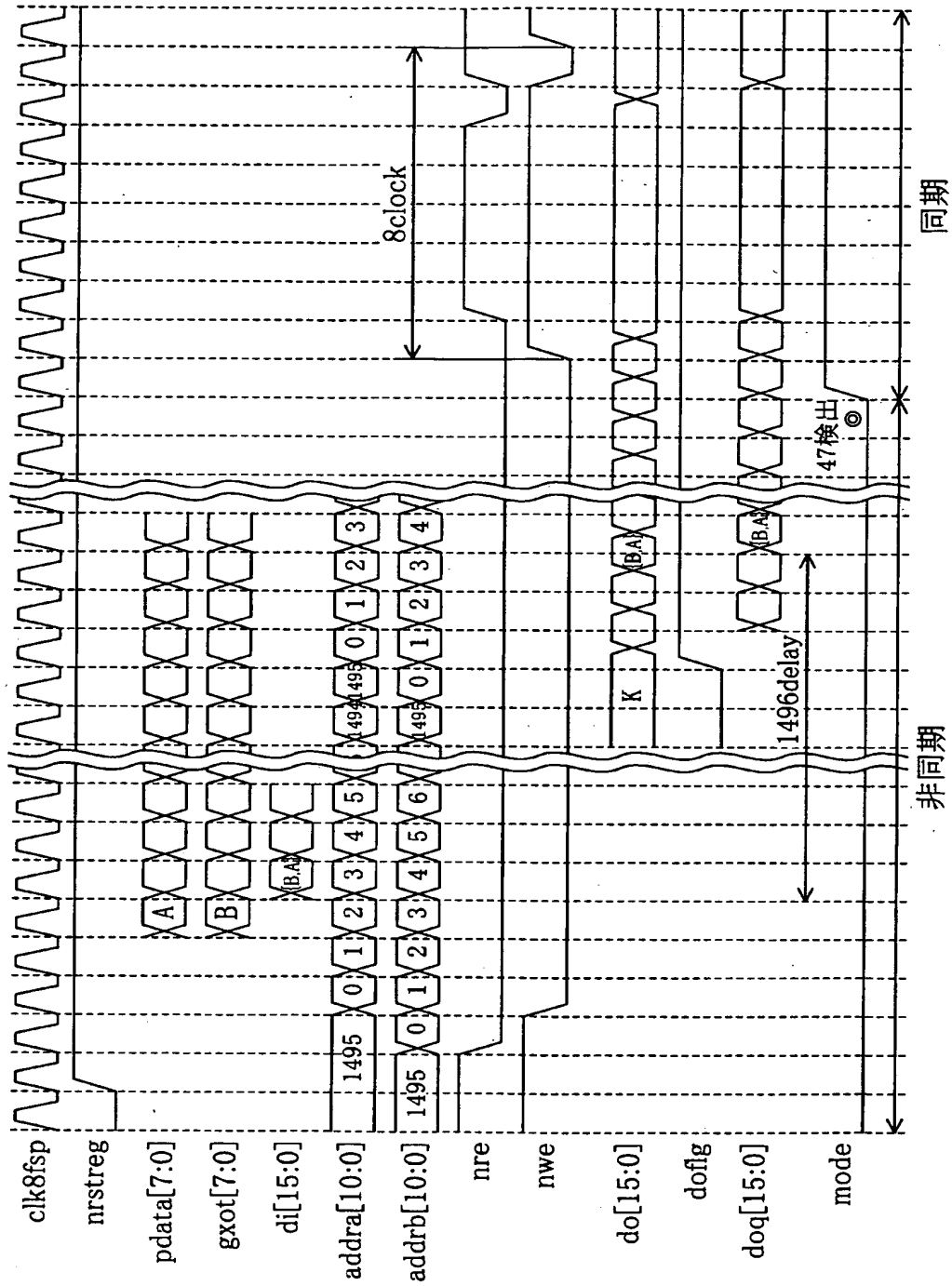


(a)

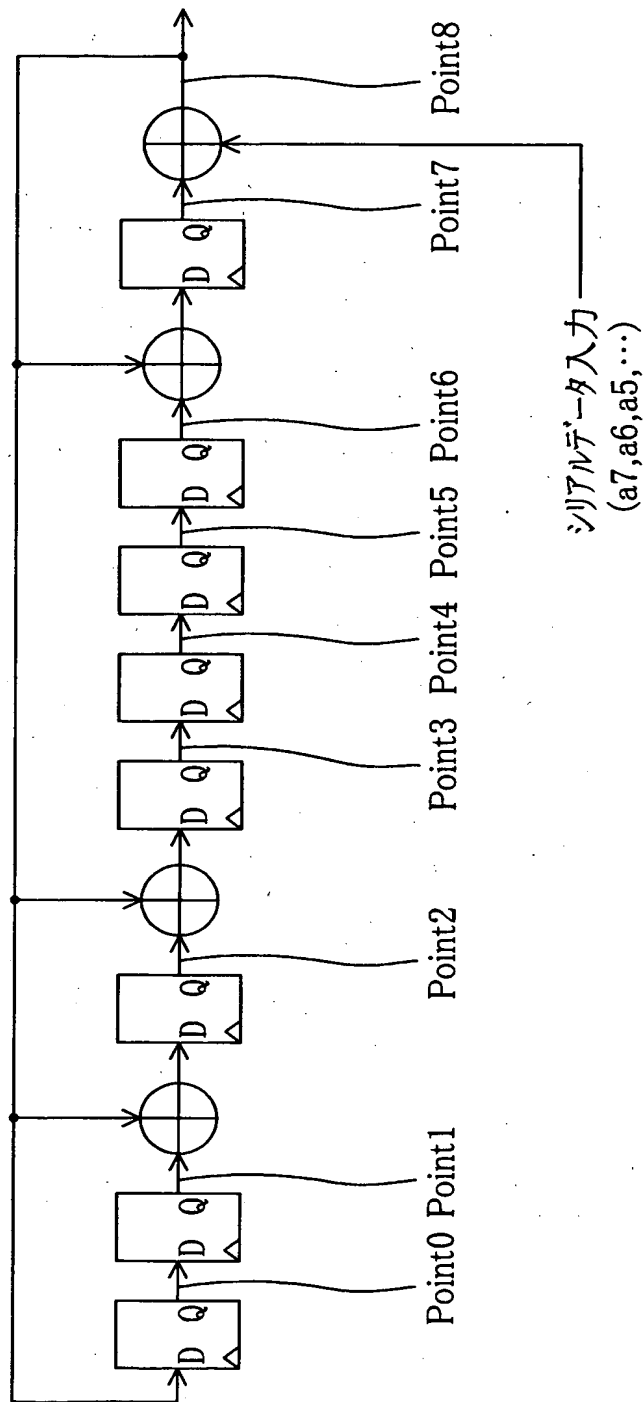
time	Read Address (addrb[10:0])	Write Address (addra[10:0])	di[15:0]	do[15:0]	nwe	nre
T1495	1494	1495	data1495	data1494		
T1494	1493	1494	data1494	data1493		
...	0	0
T2	1	2	data2	data1		
T1	0	1	data1	data0		
T0	1495	0	data0	data1495		

(b)

【図 8】



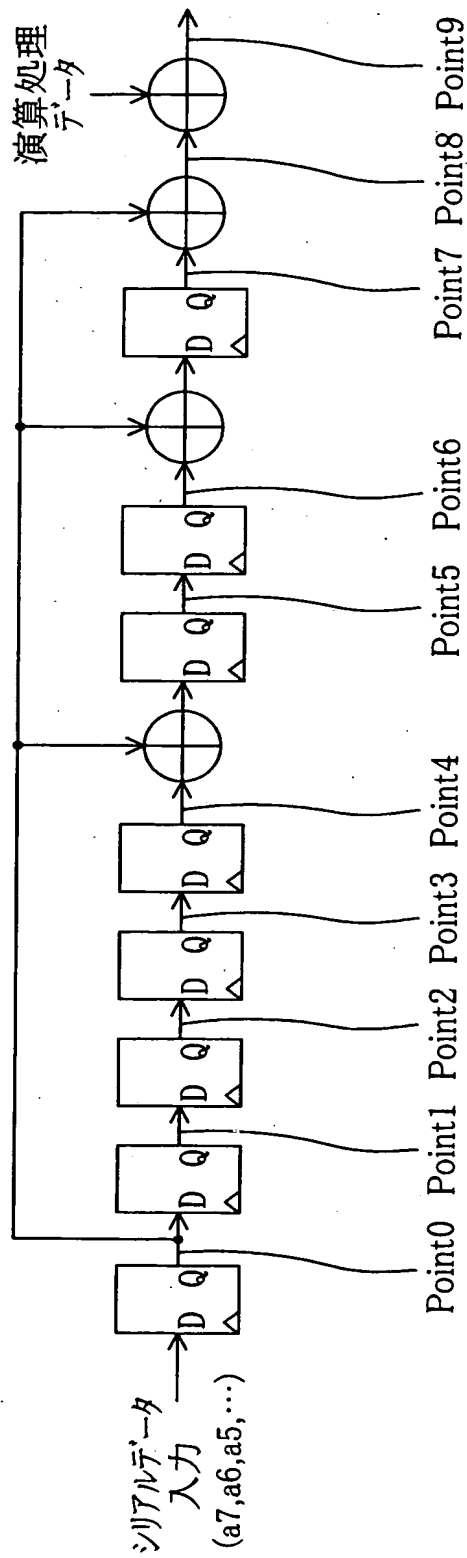
【図 9】



【図 1 0】

time	Point8	Point7	Point6
1	X7+a7	X7	X6
2	X6+X7+a7+a6	X6+X7+a7	X5
3	X5+X6+X7+a7+a6+a5	X5+X6+X7+a7+a6	X4
4	X4+X5+X6+X7+a7+a6+a5+a4	X4+X5+X6+X7+a7+a6+a5	X3
5	X3+X4+X5+X6+X7+a7+a6+a5+a4+a3	X3+X4+X5+X6+X7+a7+a6+a5+a4	X2+X7+a7
6	X2+X3+X4+X5+X6+a6+a5+a4+a3+a2	X2+X3+X4+X5+X6+a6+a5+a4+a3	X1+X6+a6
7	X1+X2+X3+X4+X5+a5+a4+a3+a2+a1	X1+X2+X3+X4+X5+a5+a4+a3+a2	X0+X5+a5
8	X0+X1+X2+X3+X4+a4+a3+a2+a1+a0	X0+X1+X2+X3+X4+a4+a3+a2+a1	X4+X7+a7+a4
time	Point5	Point4	Point3
1	X5	X4	X3
2	X4	X3	X2+X7+a7
3	X3	X2+X7+a7	X1+X6+a6
4	X2+X7+a7	X1+X6+a6	X0+X5+a5
5	X1+X6+a6	X0+X5+a5	X4+X7+a7+a4
6	X0+X5+a5	X4+X7+a7+a4	X3+X6+X7+a7+a6+a3
7	X4+X7+a7+a4	X3+X6+X7+a7+a6+a3	X2+X5+X6+a6+a5+a2
8	X3+X6+X7+a7+a6+a3	X2+X5+X6+a6+a5+a2	X1+X4+X5+X7+a7+a5+a4+a1
time	Point2	Point1	Point0
1	X2	X1	X0
2	X1+X7+a7	X0	X7+a7
3	X0+X6+X7+a7+a6	X7+a7	X6+X7+a7+a6
4	X5+X6+a6+a5	X6+X7+a7+a6	X5+X6+X7+a7+a6+a5
5	X4+X5+a5+a4	X5+X6+X7+a7+a6+a5	X4+X5+X6+X7+a7+a6+a5+a4
6	X3+X4+a4+a3	X4+X5+X6+X7+a7+a6+a5+a4	X3+X4+X5+X6+X7+a7+a6+a5+a4+a3
7	X2+X3+X7+a7+a3+a2	X3+X4+X5+X6+X7+a7+a6+a5+a4+a3	X2+X3+X4+X5+X6+a6+a5+a4+a3+a2
8	X1+X2+X6+X7+a7+a6+a2+a1	X2+X3+X4+X5+X6+a6+a5+a4+a3+a2	X1+X2+X3+X4+X5+a5+a4+a3+a2+a1

【図 1 1】

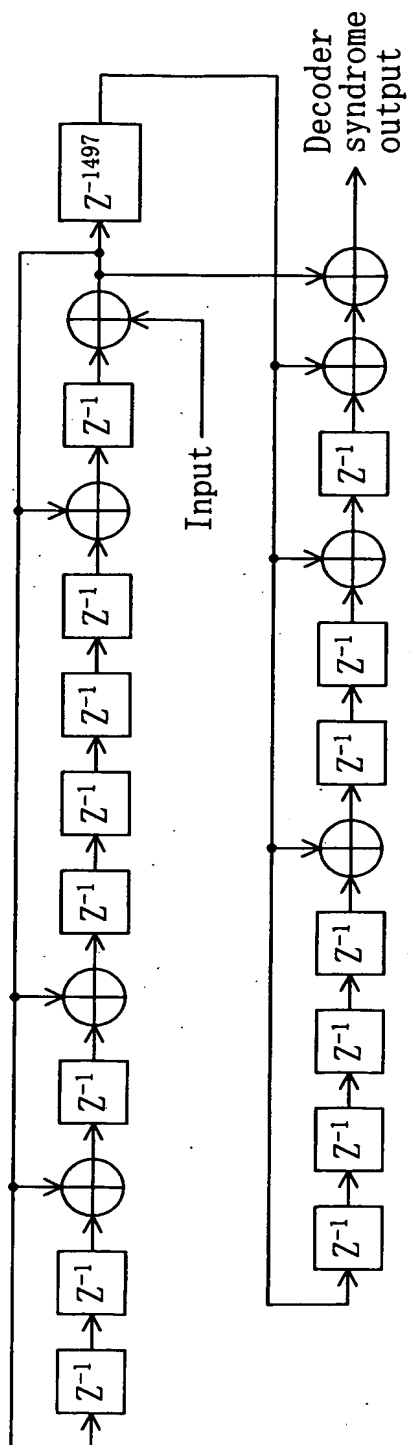


【図 1 2】

time	Point9	Point8	Point7	Point6	Point5
1	$gxot1d7+X7+a7$	$X7+a7$	$X7$	$X6$	$X5$
2	$gxot1d6+X6+X0+a7$	$X6+X0+a7$	$X6+X0$	$X5$	$X4+X0$
3	$gxot1d5+X5+a7+a6$	$X5+a7+a6$	$X5+a7$	$X4+X0$	$X3+a7$
4	$gxot1d4+X4+X0+a6+a5$	$X4+X0+a6+a5$	$X4+X0+a6$	$X3+a7$	$X2+a6$
5	$gxot1d3+X3+a7+a5+a4$	$X3+a7+a5+a4$	$X3+a7+a5$	$X2+a6$	$X1+a5$
6	$gxot1d2+X2+a6+a4+a3$	$X2+a6+a4+a3$	$X2+a6+a4$	$X1+a5$	$X0+a4$
7	$gxot1d1+X1+a5+a3+a2$	$X1+a5+a3+a2$	$X1+a5+a3$	$X0+a4$	$a7+a3$
8	$gxot1d0+X0+a4+a2+a1$	$X0+a4+a2+a1$	$X0+a4+a2$	$a7+a3$	$a6+a2$

time	Point4	Point3	Point2	Point1	Point0
1	$X4$	$X3$	$X2$	$X1$	$X0$
2	$X3$	$X2$	$X1$	$X0$	$a7$
3	$X2$	$X1$	$X0$	$a7$	$a6$
4	$X1$	$X0$	$a7$	$a6$	$a5$
5	$X0$	$a7$	$a6$	$a5$	$a4$
6	$a7$	$a6$	$a5$	$a4$	$a3$
7	$a6$	$a5$	$a4$	$a3$	$a2$
8	$a5$	$a4$	$a3$	$a2$	$a1$

【図 13】



【書類名】 要約書

【要約】

【課題】 誤り及び同期検出回路において、パラレル-シリアル変換回路、1497段の遅延器、及びシリアル-パラレル変換回路を不要にする。

【解決手段】 7ビット単位のバイトデータはデータ並び替えブロック1により8ビットで1バイトを構成するバイトデータに並び替えられる。その後、この8ビット単位のバイトデータを一貫して用いられ、この各バイトデータがRAMで構成されるデータ記憶ブロック3に記憶される。パリティチェックブロック2は、データ並び替えブロック1からのバイトデータと、データ記憶ブロック3から1496遅延した各バイトデータとを受けて、各バイトデータの同期検出演算及びパリティチェック演算を行なう。従って、バイト-バイト変換処理により、パラレル-シリアル変換回路及びシリアル-パラレル変換回路が不要になる。RAMを用いて各バイトデータを記憶するので、1496段の遅延器が不要になる。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日

[変更理由] 新規登録

住 所 大阪府門真市大字門真1006番地
氏 名 松下電器産業株式会社